

# IDENTIFIKASI PENYAKIT GAGAL GINJAL MENGGUNAKAN METODE *NEIGHBOR WEIGHTED K-NEAREST NEIGHBOR* (NWKNN)

## SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Azizul Hanifah Hadi  
NIM: 135150201111065



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

IDENTIFIKASI PENYAKIT GAGAL GINJAL MENGGUNAKAN METODE *NEIGHBOR WEIGHTED K-NEAREST NEIGHBOR (NWKNN)*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer


Disusun Oleh :  
Azizul Hanifah Hadi  
NIM: 135150201111065

Skripsi ini telah diuji dan dinyatakan lulus pada  
16 Januari 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

  
Dian Eka Ratnawati, S.Si, M.Kom  
NIP: 19730619 200212 2 001

  
Candra Dewi, S.Kom, M.Sc  
NIP: 19771114 200312 2 001

Mengetahui  
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 Januari 2018



Azizul Hanifah Hadi

NIM: 135150201111065



## KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberi petunjuk dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Identifikasi Penyakit Gagal Ginjal Menggunakan Metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN)”. Proses penulisan skripsi ini merupakan sebuah pengalaman berharga yang dapat menjadi bekal bagi penulis untuk menjadi seseorang yang terus berusaha memperbaiki diri. Dengan tersusunnya skripsi ini, penulis ingin mengucapkan terima kasih yang tak terhingga kepada:

1. Ibu Dian Eka Ratnawati, S.Si, M.Kom, dan Ibu Candra Dewi, S.Kom, M.Sc, sebagai pembimbing pertama dan pembimbing kedua yang telah memberikan bantuan dan dengan kesabaran membimbing penulis serta senantiasa memberi semangat, sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya yang memberikan penulis kesempatan menuntut ilmu di Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Dekan Fakultas Ilmu Komputer Universitas Brawijaya yang memberikan penulis kesempatan menuntut ilmu di Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Kedua orang tua penulis, Bapak Drs. Samsuhadi dan Ibu Eny Suprimi yang telah memberikan doa, dukungan, dan motivasi kepada penulis.
5. Kakak Fauziah Arbie Hadi, Setiwan Dwi Purnomo, Indah Ratnasari serta keponakan-keponakanku Rizky Aldie Milito dan Muhammad Faiz Syarif yang telah memberikan doa dan semangat kepada penulis.
6. Teman-teman seperjuangan Informatika 2013 Dennes Nur, Muhyidin Ubaidillah, Arie Kris, Aditya Orba, serta semua pihak yang telah memberikan semangat, canda tawa, serta bantuan kepada penulis dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa penulisan ini masih jauh dari sempurna, oleh karena itu penulis membuka diri untuk segala saran dan kritik yang membangun agar skripsi ini dapat bermanfaat bagi kita semua.

Malang, 22 Januari 2018

Azizul Hanifah Hadi

azizulhanifahhadi@gmail.com

## ABSTRAK

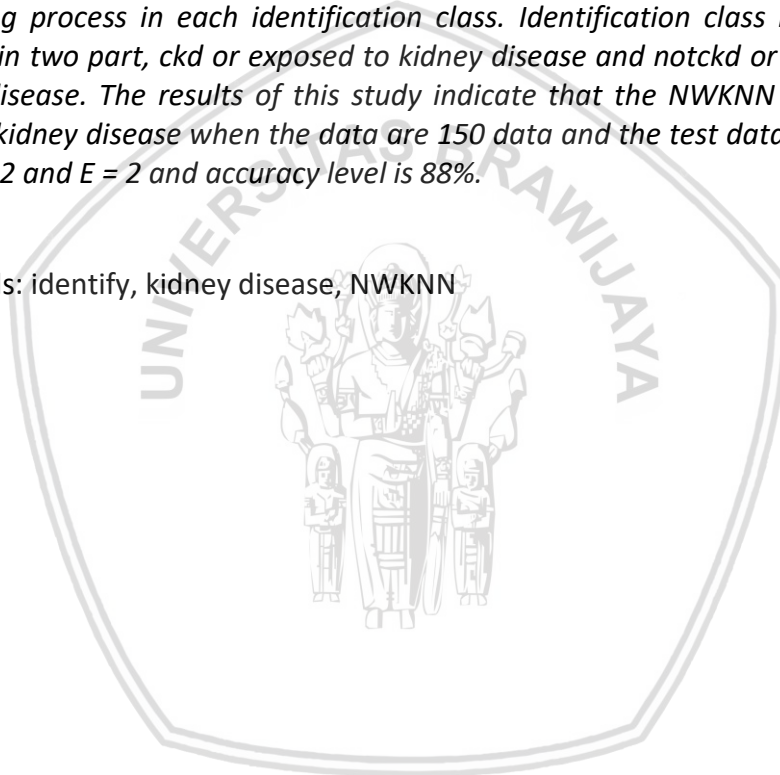
Penyakit gagal ginjal dapat disebabkan oleh beberapa faktor yakni hipertensi, kadar asam urat, kadar kreatinin, diabetes, dan masih banyak yang lain. Dari beberapa faktor itu, kita bisa mengetahui tingkat resiko terjadinya penyakit gagal ginjal. Tetapi banyak orang yang tidak menyadari, malas dan acuh mengenai pemeriksaan kesehatan terutama pada ginjal karena proses yang lama dan rumit. Menurut Indonesian Renal Registry, pada tahun 2014 pasien penyakit gagal ginjal di Indonesia mencapai 12.770 jiwa. Oleh karena itu maka dibutuhkan suatu sistem yang dapat melakukan deteksi atau identifikasi mengenai penyakit gagal ginjal. Pada penelitian ini akan dilakukan identifikasi penyakit gagal ginjal menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN). Metode ini hampir sama dengan metode KNN tetapi yang membedakan terdapat pada proses pembobotan di setiap kelas identifikasi. Kelas identifikasi pada penelitian ini ada 2 yakni *ckd* atau terkena gagal ginjal dan *notckd* atau tidak terkena gagal ginjal. Hasil dari penelitian ini menunjukkan bahwa metode NWKNN dapat melakukan identifikasi terhadap penyakit gagal ginjal pada saat data latih yang digunakan sebanyak 150 data dan data uji sebanyak 50 data dengan nilai  $K=2$  dan nilai  $E=2$  dengan hasil akurasi mencapai 88%.

Kata kunci: identifikasi, gagal ginjal, NWKNN

## ABSTRACT

*Kidney disease can be caused by several factors such as hypertension, uric acid levels, creatinine levels, diabetes, and many others. From that factors, we know about the level of kidney disease risk. Some people are unaware, lazy and indifferent about health, especially on kidney disease because of the long process and complicated. According to the Indonesian Renal Registry, in 2014 patients with kidney disease in Indonesia reach 12,770 inhabitants. Therefore, we need a system that can detect or identify the kidney disease. In this research, we will identify the kidney disease using Neighbor Weighted K-Nearest Neighbor (NWKNN) method. This method is similar to the KNN method but the differentiates are in the weighting process in each identification class. Identification class in this study decided in two part, ckd or exposed to kidney disease and notckd or not affected kidney disease. The results of this study indicate that the NWKNN method can identify kidney disease when the data are 150 data and the test data are 50 data with  $K = 2$  and  $E = 2$  and accuracy level is 88%.*

Keywords: identify, kidney disease, NWKNN



## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR SOURCE CODE .....	xii
DAFTAR PERSAMAAN .....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian pustaka .....	5
2.2 Ginjal .....	7
2.3 Penyakit Ginjal .....	7
2.3.1 Pengertian .....	7
2.3.2 Faktor pengaruh penyakit gagal ginjal .....	7
2.4 Data Tidak Seimbang .....	8
2.5 Normalisasi Data .....	9
2.6 K-Nearest Neighbor (KNN).....	9
2.7 Neighbor Weighted K-Nearest Neighbor (NWKNN).....	10
2.8 Akurasi .....	11
BAB 3 METODOLOGI .....	12

3.1 Studi literatur .....	12
3.2 Analisis kebutuhan.....	13
3.3 Pengumpulan data.....	13
3.4 Perancangan .....	13
3.5 Implementasi .....	14
3.6 Pengujian .....	14
3.7 Kesimpulan.....	14
BAB 4 PERANCANGAN.....	15
4.1 Deskripsi permasalahan.....	15
4.2 Perancangan proses.....	15
4.2.2 Proses normalisasi.....	16
4.2.3 Proses Algoritma Neighbor Weighted K-Nearest Neighbor (NWKNN).....	17
4.2.4 Proses perhitungan ketetanggaan dengan <i>Cosine Similarity (CosSim)</i> .....	18
4.2.5 Proses perhitungan bobot pada atribut <i>class</i> .....	20
4.2.6 Proses perhitungan nilai skor.....	22
4.3 Perhitungan manual .....	23
4.3.1 Normalisasi data.....	26
4.3.2 Metode NWKNN .....	27
4.4 Perancangan pengujian .....	32
4.4.1 Perancangan pengujian pengaruh nilai K dan nilai E .....	32
4.4.2 Perancangan pengujian perbandingan akurasi metode NWKNN berdasarkan pengukuran jarak .....	33
4.4.3 Perancangan pengujian perbandingan akurasi metode NWKNN dan KNN .....	33
BAB 5 IMPLEMENTASI .....	35
5.1 Implementasi program .....	35
5.1.1 Implementasi proses normalisasi data .....	35
5.1.2 Implementasi proses perhitungan kedekatan ketetanggaan .....	36
5.1.3 Implementasi proses perhitungan pembobotan kelas.....	37
5.1.4 Implementasi proses perhitungan skor .....	38
BAB 6 PENGUJIAN .....	40



6.1 Pengujian nilai K dan nilai E .....	40
6.2 Pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak .....	46
6.3 Pengujian perbandingan akurasi metode NWKNN dan KNN .....	48
BAB 7 Penutup .....	50
7.1 Kesimpulan.....	50
7.2 Saran .....	50
DAFTAR PUSTAKA.....	51
LAMPIRAN A DATA PENYAKIT GAGAL GINJAL.....	53



## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	6
Tabel 4.1 Keterangan atribut .....	24
Tabel 4.2 Sampel data latih dan data uji.....	25
Tabel 4.3 Normalisasi data.....	26
Tabel 4.4 Nilai kedekatan ketetanggaan.....	28
Tabel 4.5 Urutan kedekatan ketetanggaan.....	29
Tabel 4.6 Nilai bobot setiap jenis identifikasi .....	30
Tabel 4.7 Jarak yang telah diurutkan dengan nilai $K=5$ .....	30
Tabel 4.8 Hasil perhitungan skor.....	31
Tabel 4.9 Skenario perancangan pengujian pengaruh nilai $K$ dan nilai $E$ .....	32
Tabel 4.10 Perancangan pengujian perbandingan akurasi metode NWKNN berdasarkan pengukuran jarak .....	33
Tabel 4.11 Perancangan pengujian perbandingan akurasi metode NWKNN dan KNN .....	34
Tabel 6.1 Hasil pengujian skenario 1 nilai $E = 2$ .....	40
Tabel 6.2 Hasil pengujian skenario 2 nilai $E = 4$ .....	42
Tabel 6.3 Hasil pengujian skenario 3 nilai $E=6$ .....	43
Tabel 6.4 Hasil pengujian skenario 4 nilai $E = 8$ .....	44
Tabel 6.5 Hasil pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak.....	47
Tabel 6.6 Hasil pengujian perbandingan akurasi metode NWKNN dengan KNN .	48

## DAFTAR GAMBAR

Gambar 3.1 Diagram alir tahapan penelitian.....	12
Gambar 4.1 Diagram alir sistem.....	15
Gambar 4.2 Proses Normalisasi .....	17
Gambar 4.3 Proses NWKNN .....	18
Gambar 4.4 Proses perhitungan ketetanggaan dengan <i>cosine similarity (CosSim)</i> .....	20
Gambar 4.5 Proses perhitungan bobot.....	21
Gambar 4.6 Proses perhitungan nilai skor .....	23
Gambar 6.1 Grafik pengujian nilai K bervariasi dan nilai E=2 .....	41
Gambar 6.2 Grafik pengujian nilai K bervariasi dan nilai E=4 .....	42
Gambar 6.3 Grafik pengujian nilai K bervariasi dan nilai E=6 .....	44
Gambar 6.4 Grafik pengujian nilai K bervariasi dan nilai E=8 .....	45
Gambar 6.5 Grafik pengujian nilai E.....	46
Gambar 6.6 Grafik hasil pengujian perbandingan akurasi NWKNN berdasarkan metode pengukuran jarak.....	47
Gambar 6.7 Grafik hasil perbandingan akurasi NWKNN dan KNN .....	49

## DAFTAR SOURCE CODE

Source Code 5.1 Proses Normalisasi Data .....	35
Source Code 5.2 Proses Perhitungan Kedekatan Ketetanggaan .....	36
Source Code 5.3 Proses Perhitungan Pembobotan Kelas.....	37
Source Code 5.4 Proses Perhitungan Pembobotan Kelas.....	38



## DAFTAR PERSAMAAN

Rumus Normalisasi Data (2.1) .....	9
Rumus Euclidean Distance (2.2) .....	9
Rumus Cosine Similarity (2.3) .....	10
Perhitungan Skor Metode KNN Dengan Euclidean Distance (2.4) .....	10
Perhitungan Skor Metode KNN Dengan Cosine Similarity (2.5) .....	10
Perhitungan Bobot (2.6) .....	11
Perhitungan Skor Metode NWKNN Dengan Euclidean Distance (2.7) .....	11
Perhitungan Skor Metode NWKNN Dengan Cosine Similarity (2.8) .....	11
Perhitungan Akurasi (2.9) .....	11





## DAFTAR LAMPIRAN

LAMPIRAN A DATA PENYAKIT GAGAL GINJAL.....	53
--	----



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Ginjal merupakan dua buah organ berbentuk seperti kacang yang berada di bawah tulang rusuk dengan panjang sekitar 11 cm, lebar 6 cm, tebal 3 cm, dan berat sekitar 150 g (Waugh, et al., 2011). Ginjal mempunyai fungsi untuk menyaring atau membersihkan darah, mengatur atau mengeluarkan kelebihan cairan tubuh, sebagai penyeimbang kadar kimia darah, mengeluarkan hormon yang mengatur tekanan darah, dan lain sebagainya (Cahyaningsih, 2011). Pada intinya, ginjal merupakan organ yang sangat vital dan penting untuk menjaga kesehatan tubuh pada manusia dan jika terjadi penurunan fungsi ginjal maka dapat beresiko terjadinya penyakit ginjal (Cahyaningsih, 2011). Penyakit ginjal atau yang biasa disebut gagal ginjal adalah kelainan organ ginjal yang diakibatkan oleh berbagai faktor, misal infeksi, tumor, kelainan bawaan, dan lain-lain (Kementrian Kesehatan RI, 2013). Gagal ginjal tersebut dapat mempengaruhi fungsi serta struktur ginjal yang memiliki tingkat keparahan berbeda. Menurut Indonesian Renal Registry, pada tahun 2014 pasien penyakit gagal ginjal di Indonesia mencapai 12.770 jiwa, dan yang terbanyak disebabkan karena hipertensi. Jika pasien sudah terdiagnosis gagal ginjal, maka penanganannya adalah dengan cara melakukan hemodialisis atau cuci darah secara berkala serta bisa juga melakukan transplantasi ginjal untuk menggantikan ginjal sebelumnya yang telah rusak atau tidak berfungsi. Tetapi hemodialisis dan transplantasi ginjal membutuhkan biaya yang tidak sedikit, sehingga diperlukan deteksi atau pemeriksaan secara dini agar resiko terjadinya penyakit gagal ginjal dapat segera diatasi.

Terjadinya penyakit gagal ginjal dapat disebabkan beberapa faktor yang diantaranya adalah hipertensi, kadar asam urat, kadar kreatinin, diabetes, dan masih banyak yang lainnya. Dari beberapa faktor tersebut, kita dapat mengetahui tingkat resiko terjadinya penyakit gagal ginjal. Tetapi banyak yang tidak menyadari, malas serta acuh mengenai pemeriksaan kesehatan terutama pada ginjal karena proses yang rumit dan lama. Proses pemeriksaan kesehatan biasanya dari cek laboratorium dan selanjutnya konsultasi dokter. Tetapi jarak dari lokasi laboratorium menuju dokter spesialis atau sebaliknya biasanya cukup jauh dan terkadang konsultasi ke dokter spesialis harus diperlukan antre jadwal periksa, sehingga berakibat beberapa orang malas untuk melakukan pemeriksaan kesehatan tersebut. Jika beberapa orang mengetahui bahwa mereka belum atau tidak terkena gagal ginjal, maka hal tersebut dapat dihindari dan tidak perlu dilakukan cuci darah atau transplantasi ginjal (Maghfirah, et al., 2014). Berdasar penjelasan tersebut maka dibutuhkan suatu sistem yang dapat melakukan deteksi atau identifikasi mengenai penyakit gagal ginjal. Salah satu cara yang dapat digunakan adalah dengan menggunakan klasifikasi (Maghfirah, et al., 2014). Pada penelitian ini terdapat dua kelas dalam melakukan identifikasi yakni mengalami gagal ginjal dan tidak mengalami gagal ginjal.

Terkadang pada klasifikasi terdapat data yang tidak merata atau data yang tidak seimbang. Seperti data pada penelitian yang penulis gunakan yakni terdapat 43 data mengalami gagal ginjal dan 107 data tidak mengalami gagal ginjal. Salah satu pemecahan terkait data yang tidak seimbang adalah menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN). Penelitian menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN) dengan data yang tidak seimbang yang dilakukan oleh Arissaputra et al. (2015) mengenai klasifikasi dokumen tanaman obat memiliki nilai rata-rata *f-measure* sebesar 86,37% (Arissaputra, et al., 2015). Pengaruh data latih tidak seimbang pada penelitian ini dapat mempengaruhi nilai akurasi. Hasilnya metode NWKNN lebih baik 6,7% daripada metode KNN. Penelitian berikutnya mengenai metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN) dengan data kelas tidak seimbang dilakukan oleh Fadila et al. (2016). Hasil dari penelitian ini yakni sistem mampu melakukan klasifikasi terhadap anak usia dini yang terserang penyakit *Attention Deficit Hyperactivity Disorder*. Keakuratan pada sistem ini dapat mencapai 95% dan lebih baik 2% daripada menggunakan metode KNN (Fadila, et al., 2016). Pada penelitian ini penulis menggunakan data penyakit gagal ginjal yang beberapa kelas datanya tidak merata atau seimbang, sehingga penulis menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN).

## 1.2 Rumusan masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana menerapkan metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal?
2. Bagaimana hasil pengujian dan analisis parameter metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal?
3. Bagaimana perbandingan nilai akurasi dari sistem identifikasi penyakit gagal ginjal menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN) dan *k-nearest neighbor* (KNN)?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Menerapkan metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal.
2. Menguji dan menganalisis parameter metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal.
3. Mengukur perbandingan nilai akurasi dari sistem identifikasi penyakit gagal ginjal menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN) dan *k-nearest neighbor* (KNN).

## 1.4 Manfaat

Manfaat dari penelitian ini adalah dapat membantu seseorang untuk mengetahui tingkat resiko dari penyakit gagal ginjal serta dapat segera melakukan pengobatan jika terkena penyakit tersebut.

## 1.5 Batasan masalah

Pada penelitian ini mempunyai beberapa batasan masalah yaitu:

1. Data penelitian sebanyak 200 data dan diperoleh dari *UCI Machine Learning Repository: Chronic Kidney Disease*.
2. Terdapat 24 atribut yang digunakan untuk identifikasi yakni meliputi *age, blood pressure, specific gravity, albumin, sugar, red blood cells, pus cell, pus cell clumps, bacteria, blood glucose random, blood urea, serum creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count, red blood cell count, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema*, dan *anemia*.
3. Setiap atribut tidak ada perbedaan bobot.
4. Terdapat *missing value*/data yang hilang pada beberapa data uji dan nantinya akan diisi dengan menggunakan metode KNN melalui perhitungan manual.
5. Keluaran dari sistem berupa hasil identifikasi mengalami gagal ginjal atau tidak mengalami gagal ginjal.

## 1.6 Sistematika pembahasan

Sistematika penyusunan laporan ditujukan untuk memberikan gambaran dan uraian dari penyusunan skripsi ini secara garis besar yang meliputi beberapa bab, sebagai berikut:

### BAB 1 PENDAHULUAN

Bab ini berisi mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika yang terkait dengan identifikasi penyakit gagal ginjal menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN).

### BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menjelaskan tentang kajian pustaka atau penjelasan tentang penelitian sebelumnya, penjelasan tentang gagal ginjal dan metode *neighbor weighted k-nearest neighbor* (NWKNN).

### BAB 3 METODOLOGI

Bab ini menjelaskan langkah yang selanjutnya dilakukan dalam penelitian yang meliputi studi literatur, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi, pengujian dan analisis serta kesimpulan.

**BAB 4 PERANCANGAN**

Bab ini membahas tentang analisis serta perancangan yang akan dibangun dalam implementasi metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal, seperti perancangan algoritma, perancangan manualisasi dan alur program yang selanjutnya akan diimplementasikan.

**BAB 5 IMPLEMENTASI**

Bab ini membahas tentang implementasi dari metode yang digunakan yaitu metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal yang sesuai dengan perancangan sistem yang telah dibuat.

**BAB 6 PENGUJIAN**

Bab ini menguraikan jawaban atas pertanyaan pada implementasi metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal serta pengujian dari metode yang digunakan.

**BAB 7 PENUTUP**

Bab ini berisikan kesimpulan berdasarkan hasil dari pengujian dan analisa metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal dan menyertakan saran yang dapat digunakan untuk pengembangan selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tentang dasar teori yang berkaitan dengan penelitian, seperti objek penelitian dan metode yang digunakan untuk menunjang topik pembahasan.

### 2.1 Kajian pustaka

Dalam penyusunan penelitian ini, digunakan penelitian yang memiliki kemiripan dengan penelitian sebelumnya. Terdapat beberapa penelitian yang menjadi referensi dalam kajian pustaka ini. Pada penelitian sebelumnya yang menjadi referensi berkaitan dengan objek tentang gagal ginjal dan penelitian dengan menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN) dengan objek yang berbeda.

Penelitian pertama dari Maghfirah et al. (2014) dengan objek penyakit gagal ginjal. Pada penelitian ini, penulis meneliti tentang tingkat resiko penyakit gagal ginjal di Probolinggo, Jawa Timur. Penulis menggunakan 6 parameter untuk dapat membantu dalam melakukan prediksi tingkat resiko penyakit gagal ginjal yaitu antara lain umur, kadar urea, kadar BUN, kadar kreatinin, kadar uric acid dan kadar trigliserida. Dari parameter tersebut nantinya akan diperoleh kesimpulan berupa tingkat resiko. Tingkat resiko pada penelitian ini dibagi menjadi 3 yaitu normal, sedang dan tinggi. Data yang digunakan sebanyak 300 yang mana 150 data penyakit gagal ginjal untuk laki-laki dan 150 data penyakit gagal ginjal untuk perempuan. Hasil akhir dari penelitian ini memiliki nilai akurasi sebesar 80%.

Penelitian berikutnya yakni mengenai metode *neighbor weighted k-nearest neighbor* (NWKNN) yang dilakukan oleh Arissaputra et al. (2015). Penelitian ini memiliki tujuan untuk mengklasifikasikan data tanaman obat berdasarkan kategorinya. Data yang digunakan terdiri atas 14 kategori/famili dengan jumlah data dari setiap kategorinya berbeda-beda. Metode NWKNN dapat digunakan untuk mengklasifikasikan dokumen tanaman obat dengan baik pada nilai  $k=5$  dengan rata-rata *f-measure* sebesar 86,37% (Arissaputra, et al., 2015). Pengaruh data latih tidak seimbang pada penelitian ini dapat mempengaruhi nilai akurasi. Hasilnya metode NWKNN lebih baik 6,7% dari pada metode KNN.

Penelitian selanjutnya mengenai identifikasi jenis *attention deficit hyperactivity disorder* (ADHD) pada anak usia dini oleh Fadila et al. (2016). Penelitian ini bertujuan untuk dapat mengidentifikasi jenis Attention Deficit Hyperactivity Disorder (ADHD) pada anak usia dini dengan mengimplementasikan metode NWKNN. Pada penelitian ini terdapat perbedaan jumlah data di setiap kelas sehingga terlihat adanya kelas mayoritas dan kelas minoritas. Metode NWKNN dapat melakukan identifikasi jenis ADHD dengan nilai akurasi sebesar 95% pada data latih sebanyak 80, nilai  $k=10$  dan nilai eksponen=4. Pada penelitian ini juga menunjukkan bahwa NWKNN memberikan nilai rata-rata akurasi sebesar 2% lebih baik dari pada metode KNN (Fadila, et al., 2016).

Menurut penjelasan penelitian-penelitian sebelumnya, maka penulis akan menyelesaikan permasalahan mengenai identifikasi penyakit gagal ginjal menggunakan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Penelitian ini menggunakan data dari *UCI Machine Learning Repository: Chronic Kidney Disease* dengan 200 data penyakit gagal ginjal, 24 atribut dan 2 kelas akhir identifikasi.

**Tabel 2.1 Kajian Pustaka**

No	Penulis	Objek	Metode	Hasil
1	Maghfirah, et al., 2014	Tingkat resiko penyakit gagal ginjal	<i>Fuzzy K-Nearest Neighbor (F-KNN)</i>	Penelitian ini dapat melakukan prediksi untuk tingkat resiko penyakit gagal ginjal menggunakan metode F-KNN serta memiliki nilai akurasi sebesar 80%
2	Arissaputra, et al., 2015	Dokumen tanaman obat	<i>Neighbor Weighted K-Nearest Neighbor (NWKNN)</i>	Metode NWKNN dapat digunakan untuk mengklasifikasikan dokumen tanaman obat dengan baik pada nilai $k=5$ dengan rata-rata <i>f-measure</i> sebesar 86,37%. Pengaruh data tidak seimbang pada penelitian ini dapat mempengaruhi nilai akurasi. Hasilnya lebih baik 6,7% dari pada metode KNN
3	Fadila, et al., 2016	Jenis Attention Deficit Hyperactivity Disorder (ADHD) Pada Anak Usia Dini	<i>Neighbor Weighted K-Nearest Neighbor (NWKNN)</i>	Metode NWKNN dapat melakukan identifikasi jenis ADHD dengan nilai akurasi sebesar 95% pada data latih sebanyak 80, nilai $k=10$ dan nilai eksponen=4. Pada penelitian ini menunjukkan bahwa NWKNN memberikan nilai rata-rata akurasi sebesar 2% lebih baik dari pada metode KNN

No	Penulis	Objek	Metode	Hasil
4	Azizul, 2017	Penyakit gagal ginjal	<i>Neighbor Weighted K-Nearest Neighbor (NWKNN)</i>	-

## 2.2 Ginjal

Ginjal merupakan organ berbentuk kacang, yang panjangnya sekitar 11 cm, lebar 6 cm, tebal 3 cm, serta beratnya 150 g (Waugh, et al., 2011). Ginjal mempunyai peran yang sangat penting dalam menjaga kesehatan tubuh secara menyeluruh karena ginjal adalah salah satu organ vital dalam tubuh. Bila ginjal tidak bekerja sebagaimana mestinya maka akan timbul masalah kesehatan yang berkaitan dengan penyakit ginjal kronik (Cahyaningsih, 2011). Kerja ginjal yang berfungsi sebagai mencuci atau membersihkan darah begitu efektif serta efisien. Maksud dari efektif yakni pada saat menentukan zat yang dibuang dari tubuh dan maksud dari efisien adalah penggunaan waktu, energi serta ruang untuk menjalankan fungsi ginjal tersebut (Indra, 2006). Fungsi ginjal antara lain (Cahyaningsih, 2011; Indra, 2006):

1. Membersihkan darah dan mengeluarkan kelebihan cairan tubuh,
2. Mengatur keseimbangan kadar kimia darah dalam tubuh,
3. Mengeluarkan hormon yang mengatur tekanan dan aliran darah.

## 2.3 Penyakit Ginjal

### 2.3.1 Pengertian

Penyakit ginjal atau yang biasa disebut gagal ginjal adalah kelainan organ ginjal yang diakibatkan oleh berbagai faktor, misal infeksi, tumor, kelainan bawaan, dan lain-lain (Kementrian Kesehatan RI, 2013). Gagal ginjal adalah salah satu penyakit mematikan dan penyebab kematian utama di dunia (Indra, 2006). Ginjal yang sakit ditandai dengan pengeluaran kotaran atau racun dan pengaturan air, keseimbangan asam-basa yang terganggu, tekanan darah tinggi, anemia dan sakit tulang (Budyanto, 2002). Penyakit pada ginjal yang berat dibagi dua jenis atau kategori yakni gagal ginjal akut dan gagal ginjal kronis. Gagal ginjal akut dapat ditandai dengan fungsi ginjal berhenti secara mendadak yang dapat terjadi pemulihan secara berangsur-angsur (Indra, 2006). Sedangkan gagal ginjal kronis ditandai dengan hilangnya proses dari fungsi nefron secara progresif sehingga penurunan fungsi ginjal menurun secara bertahap.

### 2.3.2 Faktor pengaruh penyakit gagal ginjal

Beberapa faktor pengaruh yang dapat menyebabkan terjadinya penyakit gagal ginjal adalah sebagai berikut (Rubini & Eswaran, 2015):

1. *Age*: umur,
2. *Blood Pressure*: tekanan darah,
3. *Specific Gravity*: berat jenis minyak,
4. *Albumin*: sebagai pengatur tekanan osmotik dalam darah,
5. *Sugar*: gula darah,
6. *Red Blood Cells*: sel darah merah,
7. *Pus Cell*: suatu cairan dari hasil proses peradangan yang terbentuk dari sel-sel leukosit(nanah),
8. *Pus Cell Clumps*: gumpalan nanah,
9. *Bacteria*: bakteri pada urin,
10. *Blood Glucose Random*: gula darah sewaktu,
11. *Blood Urea*: kadar urea pada darah,
12. *Serum Creatinine*: asam amino yang diproduksi oleh hati, pancreas dan ginjal,
13. *Sodium*: natrium,
14. *Potassium*: kalium,
15. Hemoglobin: indeks kapasitas pembawa oksigen pada darah,
16. *Packed Cell Volume*: proporsi volume darah yang ditempati oleh eritrosit (rasio sel darah merah ke seluruh volume darah),
17. *White Blood Cell Count*: mengukur atau menentukan jumlah dan jenis leukosit yang ada dalam sampel darah,
18. *Red Blood Cell Count*: mengukur atau menentukan jumlah dan jenis eritrosit yang ada dalam sampel darah,
19. *Hypertention*: tekanan darah tinggi(hipertensi),
20. *Diabetes Mellitus*: kencing manis,
21. *Coronary Artery Disease*: penyakit jantung coroner,
22. *Appetite*: nafsu makan,
23. *Pedal Edema*: pengumpulan cairan secara abnormal pada kaki,
24. Anemia: keadaan ketika jumlah sel darah merah atau konsentrasi pengangkut oksigen dalam darah (Hb) tidak mencukupi untuk kebutuhan fisiologis tubuh.

## 2.4 Data Tidak Seimbang

Pada suatu penelitian, adanya suatu data pada suatu kelas tidak sama, ada yang memiliki jumlah data yang banyak dibanding kelas lainnya. Kondisi tersebut dapat dikatakan sebagai kondisi data tidak seimbang. Data tidak seimbang adalah kondisi dimana persebaran data tidak merata di dalam kelas yang ada. Pada data

yang tidak seimbang akan ada istilah yakni kelas mayoritas dan kelas minoritas (Ridok & Latifah, 2015).

## 2.5 Normalisasi Data

Variabel atau atribut adalah bagian dari data yang mewakili karakteristik dari objek data. Nilai ini biasanya memiliki rentang yang hampir sama dan ada pula yang bervariasi. Misal atribut A memiliki nilai rentang besar yakni 1000-2000 sedangkan atribut B bernilai hanya 1 atau 0, sehingga selisih jarak kedua atribut tersebut tidak seimbang. Maka diperlukan normalisasi untuk menjaga keseimbangan data. Salah satu teknik normalisasi data yakni normalisasi *min-max*. Berikut merupakan rumus penggunaan teknik normalisasi *min-max* (Jain & Bhandare, 2011):

$$V' = \frac{V - \min_A}{\max_A - \min_A} \quad (2.1)$$

Dimana:

$V'$  = hasil perhitungan normalisasi *min-max*

$V$  = nilai atribut A yang akan dinormalisasi

$\min_A$  = nilai minimum dari suatu atribut A

$\max_A$  = nilai maksimum dari suatu atribut A

## 2.6 K-Nearest Neighbor (KNN)

Metode KNN merupakan salah satu metode klasifikasi yang didasarkan atas kedekatan ketetanggaan nilai kelas, klasifikasi dilakukan berdasarkan kesamaan atau jarak dari suatu data dengan data yang lain. Dalam KNN, untuk menghitung nilai kedekatan ketetanggaan antara data uji dengan data latih dapat menggunakan *Euclidean Distance* atau *Cosine Similarity* (*CosSim*).

Langkah-langkah pada algoritma KNN adalah sebagai berikut (Fadila, et al., 2016):

1. Menentukan nilai K tetangga.
2. Menghitung nilai kedekatan ketetanggaan antara data uji dan data latih (menggunakan *Euclidean Distance* atau *Cosine Similarity*/*CosSim*).
  - Untuk menghitung nilai kedekatan ketetanggaan menggunakan rumus *Euclidean Distance* seperti berikut:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2} \quad (2.2)$$

Dimana:

$X_1$  = nilai tiap data latih

$X_2$  = nilai tiap data uji

$n$  = banyak data



$i$  = data ke- $i$

- Untuk menghitung nilai kedekatan ketetanggaan menggunakan rumus *Cosine Similarity (CosSim)* seperti berikut:

$$\text{CosSim}(q, d_j) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^m (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^m w_{ij}^2 \cdot \sum_{i=1}^m w_{iq}^2}} \quad (2.3)$$

Dimana:

$Q$  = data uji

$d_j$  = data latih

$\vec{d_j} \cdot \vec{q}$  = hasil total perkalian vektor antara data latih dengan data uji

$|\vec{d_j}| \cdot |\vec{q}|$  = hasil total perkalian vektor antara norm data latih dengan data uji

$w_{ij}$  = bobot nilai / pada data latih  $j$

$w_{iq}$  = bobot nilai / pada data uji

$m$  = banyak jumlah nilai

3. Mengurutkan hasil perhitungan kedekatan ketetanggaan berdasar jarak.
4. Mengumpulkan kategori klasifikasi *nearest neighbor*.
5. Menggunakan kategori *nearest neighbor* paling mayoritas untuk menghitung nilai skor. Nilai skor digunakan sebagai acuan klasifikasi.

- Perhitungan skor pada metode KNN dengan nilai kedekatan ketetanggaan menggunakan perhitungan *Euclidean Distance*:

$$\text{Skor}(X, C_i) = (\sum_{d_j \in \text{KNN}(X)} ((\sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2}) * \delta(d_j, C_i))) \quad (2.4)$$

Dimana:

$d_j \in \text{KNN}(x)$  = data latih  $d_j$  pada kumpulan tetangga terdekat dari data uji  $x$

$\delta(d_j, C_i)$  = akan bernilai 1 jika jarak  $\in C_i$  dan bernilai 0 jika nilai jarak  $\notin C_i$

$C_i$  = jenis atau kelas  $i$

- Perhitungan skor pada metode KNN dengan nilai kedekatan ketetanggaan menggunakan perhitungan *Cosine Similarity (CosSim)*:

$$\text{Skor}(X, C_i) = \sum_{d_j \in \text{KNN}(X)} ((\text{Sim}(q, d_j) * \delta(d_j, C_i))) \quad (2.5)$$

Dimana:

$\text{Sim}(q, d_j)$  = nilai *CosSim* antara data uji dan data latih

## 2.7 Neighbor Weighted K-Nearest Neighbor (NWKNN)

Algoritma NWKNN sama seperti algoritma KNN, tetapi perbedaan terdapat pada pemberian bobot yang dilakukan algoritma NWKNN (Feizar, et al., 2014). Pada tetangga yang berasal dari kelas mayoritas maka diberi bobot yang kecil,

sedangkan pada tetangga yang berasal dari kelas minoritas maka diberi bobot yang besar (Ridok & Latifah, 2015). Perhitungan bobot dapat dilakukan dengan persamaan (Tan, 2005):

$$Weight_i = \frac{1}{\left( \frac{Num(c_i^d)}{\min\{Num(c_j^d) | n = 1, \dots, K^*\}} \right)^{\frac{1}{exponen}}} \quad (2.6)$$

Dimana:

$Num(C_i^d)$  = banyak data latih  $d$  pada kelas  $i$

$Num(C_j^d)$  = banyak data latih  $d$  pada kelas  $j$ , dimana  $j$  terdapat dalam himpunan  $k$  tetangga terdekat

$Exponen$  = nilai  $exponen$  lebih dari 1

Setelah mendapat nilai bobot, maka menghitung nilai skor dengan cara mengalikan setiap jumlah skor dari tiap kelas dengan masing-masing bobot dari tiap kelas. Perhitungan skor pada metode NWKNN adalah sebagai berikut:

$$Skor(X, C_i) = Weight_i * (\sum_{d \in NWKNN(X)} ((\sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2}) * \delta(d_j, C_i))) \quad (2.7)$$

atau

$$Skor(X, C_i) = Weight_i * (\sum_{d \in NWKNN(X)} ((Sim(q, d_j) * \delta(d_j, C_i))) \quad (2.8)$$

Dimana:

$Weight_i$  = bobot kelas  $i$

$d \in NWKNN(x)$  = data latih  $d_j$  pada kumpulan tetangga terdekat dari data uji  $x$

$\delta(d_j, C_i)$  = akan bernilai 1 jika jarak  $\in C_i$  dan bernilai 0 jika nilai jarak  $\notin C_i$

$Sim(q, d_j)$  = nilai  $CosSim$  antara data uji dan data latih

$C_i$  = jenis atau kelas  $i$

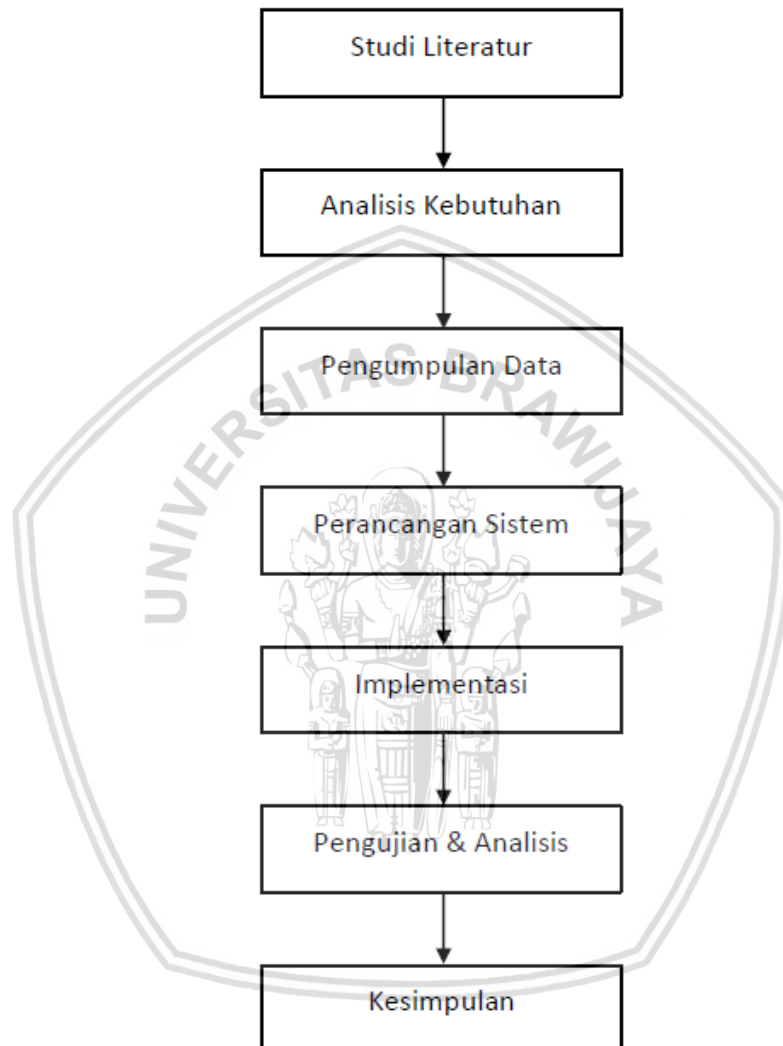
## 2.8 Akurasi

Akurasi adalah cara untuk dapat mengetahui seberapa dekat suatu angka hasil pengukuran terhadap angka yang sebenarnya. Akurasi dapat dihitung melalui persentase kebenaran dimana perbandingan antara jumlah identifikasi jenis yang benar dengan jumlah data keseluruhan. Akurasi dapat dinyatakan dalam persamaan berikut (Fadila, et al., 2016):

$$Akurasi = \frac{\text{jumlah identifikasi benar}}{\text{jumlah data}} \times 100\% \quad (2.9)$$

## BAB 3 METODOLOGI

Pada bab ini menjelaskan langkah penyelesaian masalah di dalam penelitian. Metodologi penelitian yang dilakukan pada penelitian ini terdiri dari: studi literatur, analisa kebutuhan, pengumpulan data, perancangan sistem, implementasi, pengujian, analisis dan kesimpulan.



Gambar 3.1 Diagram alir tahapan penelitian

### 3.1 Studi literatur

Di tahap ini dilakukan pembelajaran atau pustaka di bidang ilmu yang ada hubungannya dengan identifikasi penyakit gagal ginjal menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN), yaitu diantaranya:

- Metode *neighbor weighted k-nearest neighbor* (NWKNN)
- Penyakit gagal ginjal

Literatur diperoleh dari buku, jurnal, dan penelitian yang pernah dilaksanakan sebelumnya.

### 3.2 Analisis kebutuhan

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi apa saja yang dibutuhkan di dalam pembangunan dari sistem ini. Berikut merupakan kebutuhan yang diperlukan di dalam pembangunan sistem:

- a. Spesifikasi perangkat keras
  - Laptop Samsung, AMD Vision A6
  - RAM 6 GB
  - AMD Radeon Graphics 2.5 GB
- b. Spesifikasi perangkat lunak
  - Sistem Operasi Windows 10
  - Netbeans IDE 8.0
  - Sublime Text 3
  - XAMPP v3.2.1
  - Browser Google Chrome dan Microsoft Edge

### 3.3 Pengumpulan data

Pada penelitian ini menggunakan data dari UCI Machine Learning Repository: Chronic Kidney Disease. Pada data tersebut terdapat sebanyak 200 data dengan 24 atribut yakni meliputi *age, blood pressure, specific gravity, albumin, sugar, red blood cells, pus cell, pus cell clumps, bacteria, blood glucose random, blood urea, serum creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count, red blood cell count, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema, dan anemia*. Pada data yang akan di ujikan terdapat *missing value* yang mana nilai dari data *missing* tersebut sudah dicari menggunakan perhitungan manual.

### 3.4 Perancangan

Perancangan sistem dimaksud agar mempermudah peneliti pada tahapan selanjutnya yakni implementasi. Berikut ini merupakan penjelasan yang dilakukan dalam perancangan sistem:

1. Perancangan manualisasi dalam penyelesaian permasalahan dilakukan dengan membuat studi kasus perhitungan sederhana menggunakan data penyakit gagal ginjal dan metode *neighbor weighted k-nearest neighbor* (NWKNN). Kemudian didapatkan permasalahan pada penelitian ini serta solusi berupa langkah-langkah penyelesaian objek berupa algoritma dan diagram alir.

2. Perancangan pengujian terdiri dari pengujian metode dan data yang digunakan, dengan cara melakukan pengujian tingkat akurasi algoritma.

### 3.5 Implementasi

Pada tahap ini akan menjelaskan tentang implementasi metode *neighbor weighted k-nearest neighbor* (NWKNN) untuk identifikasi penyakit gagal ginjal berdasarkan perancangan manualisasi yang telah dibuat sebelumnya. Implementasi sistem menggunakan bahasa pemrograman Java dengan bantuan perangkat lunak yang telah dijelaskan pada sub bab analisis kebutuhan.

### 3.6 Pengujian

Pada tahap ini dilakukan pengujian yang bertujuan untuk menunjukkan bahwa sistem dapat bekerja dengan baik sesuai yang diharapkan oleh peneliti. Pengujian yang dapat dilakukan antara lain adalah:

1. Pengujian nilai K dan nilai E terhadap akurasi.
2. Pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak.
3. Pengujian perbandingan akurasi metode NWKNN dan KNN.

### 3.7 Kesimpulan

Pengambilan kesimpulan ini dilaksanakan setelah semua tahapan telah selesai. Pengambilan keputusan disini adalah untuk menjawab pertanyaan dari rumusan masalah yang sudah dirumuskan. Dimana untuk menjawab rumusan masalah didasarkan pada hasil implementasi serta pengujian. Tahap terakhir yaitu saran yang memiliki tujuan untuk memperbaiki jika terdapat kesalahan dan untuk memberikan pertimbangan selanjutnya.



## BAB 4 PERANCANGAN

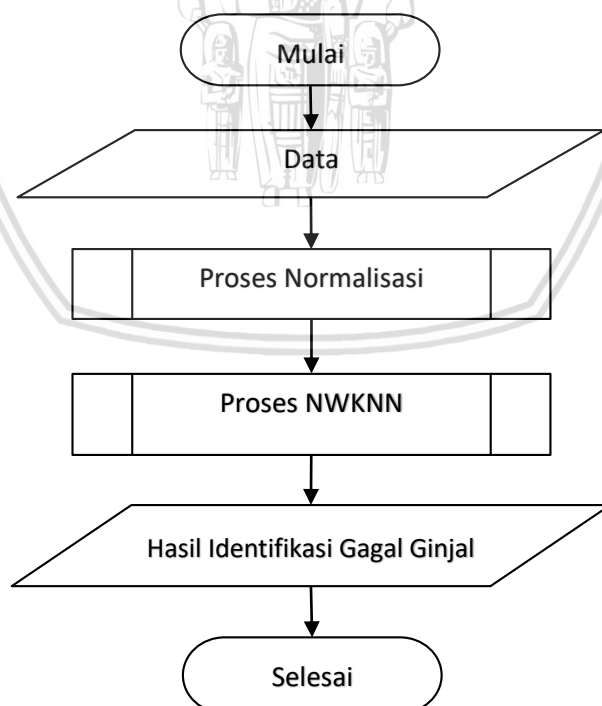
Pada tahap ini akan dilakukan proses perancangan dari identifikasi penyakit gagal ginjal menggunakan metode NWKNN. Proses perancangan terdiri dari deskripsi permasalahan, perancangan proses, perhitungan manual NWKNN dan perancangan pengujian.

### 4.1 Deskripsi permasalahan

Pada penelitian ini bertujuan untuk mengidentifikasi penyakit gagal ginjal dengan mengimplementasikan metode NWKNN dan mengetahui hasil dari metode tersebut. Terdapat 2 kelas atau hasil akhir pada penelitian ini yaitu teridentifikasi gagal ginjal (*ckd*) dan tidak teridentifikasi gagal ginjal (*notckd*). Data yang digunakan sebanyak 200 data dan 24 atribut. Untuk data latih menggunakan 150 data dan untuk data uji sebanyak 50 data.

### 4.2 Perancangan proses

Proses identifikasi penyakit gagal ginjal dilakukan dengan memasukkan nilai dari setiap gejala. Setelah memasukkan nilai di tiap gejala, maka sistem akan melakukan perhitungan. Pertama yakni proses normalisasi data latih dan data uji. Selanjutnya adalah proses perhitungan menggunakan NWKNN. Setelah melakukan perhitungan maka akan muncul hasil identifikasinya. Proses dari identifikasi penyakit gagal ginjal akan disajikan pada diagram alir pada Gambar 4.1.



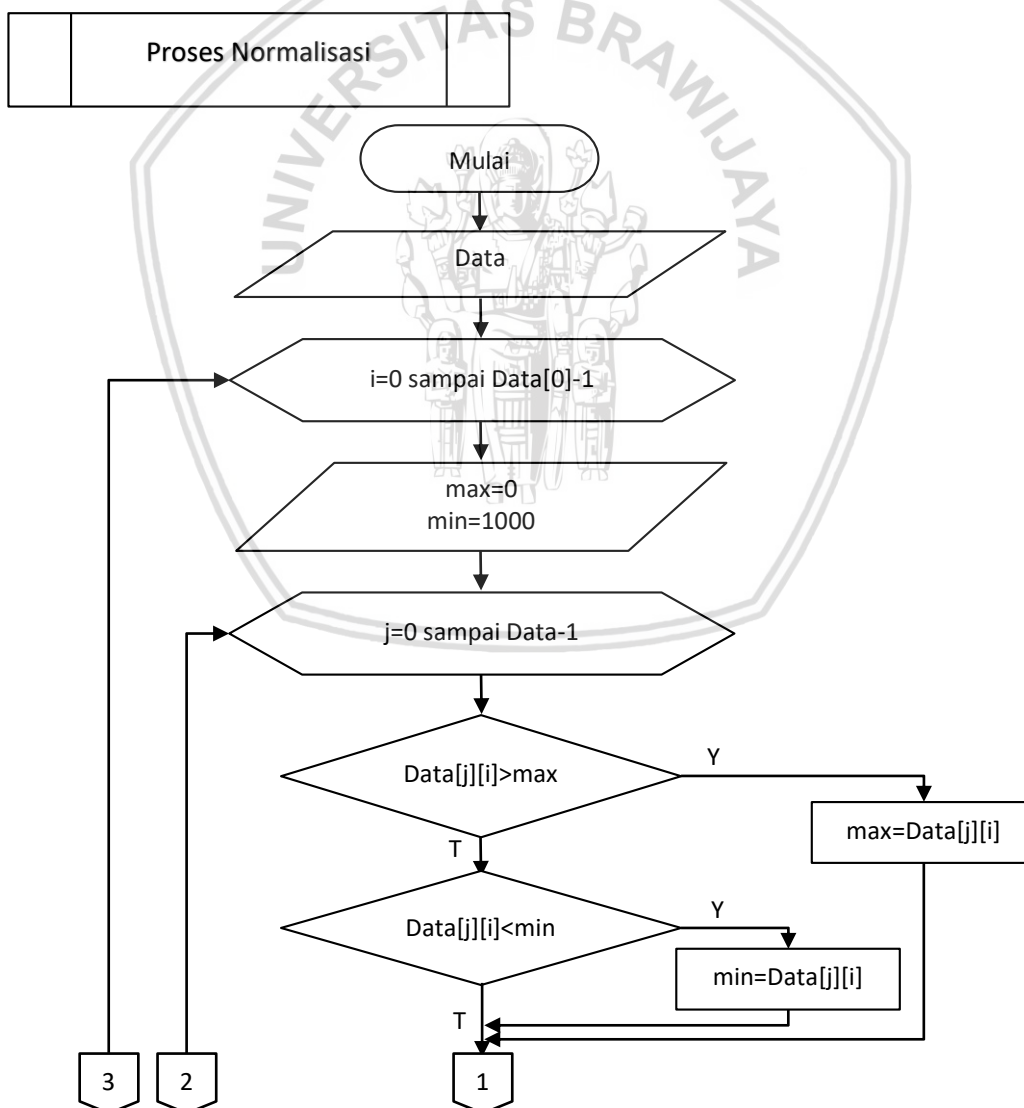
Gambar 4.1 Diagram alir sistem

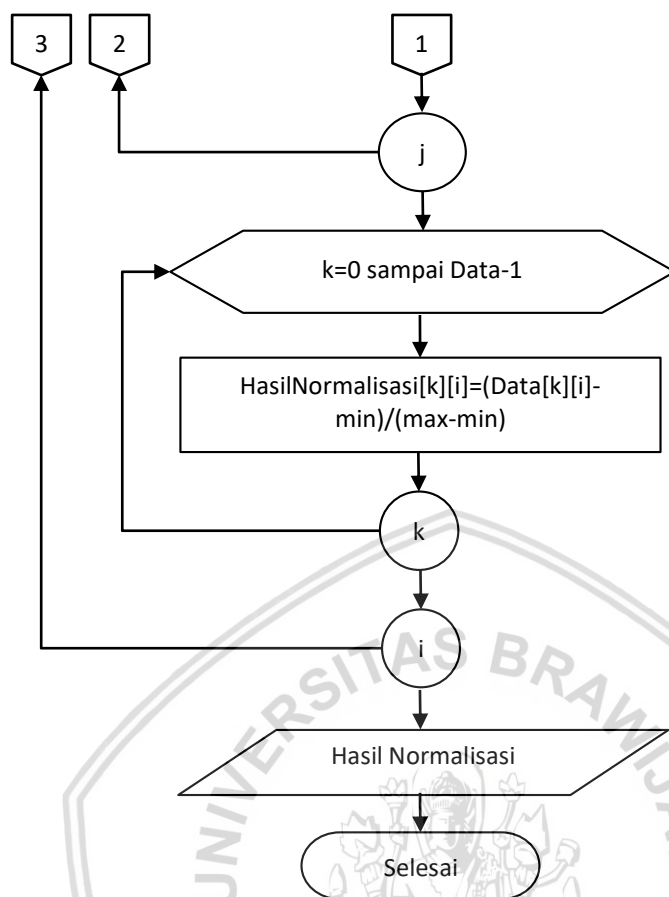
Langkah-langkah proses identifikasi gagal ginjal pada Gambar 4.1 adalah sebagai berikut:

1. Sistem menerima masukkan berupa data penyakit ginjal.
2. Melakukan proses normalisasi karena jarak nilai data yang berbeda.
3. Melakukan perhitungan dengan menggunakan NWKNN.
4. Didapatkan hasil identifikasi berupa terkena gagal ginjal atau tidak terkena gagal ginjal.

#### 4.2.2 Proses normalisasi

Proses normalisasi adalah proses transformasi data dengan tidak menghilangkan isi, dimana sebuah numerik pada suatu atribut akan ditransformasikan dalam rentang nilai yang lebih kecil seperti 0 sampai 1. Proses normalisasi yang digunakan pada penelitian ini adalah normalisasi *min-max*. Langkah dari proses normalisasi dapat dilihat pada Gambar 4.2.





**Gambar 4.2 Proses Normalisasi**

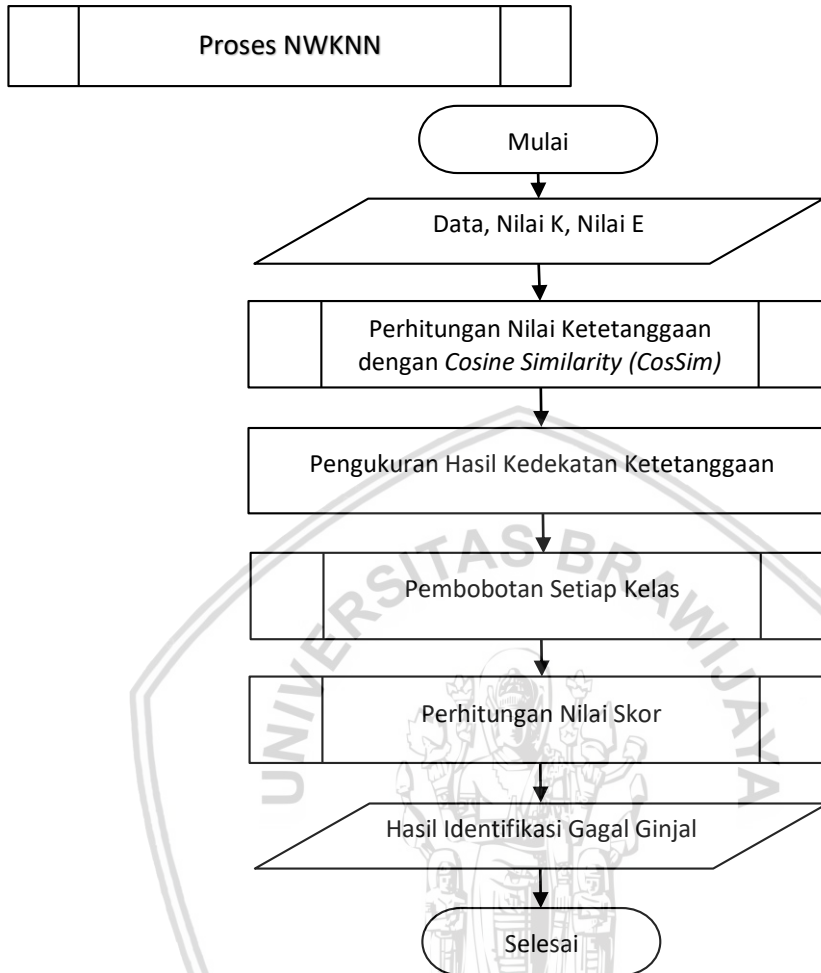
Langkah-langkah proses normalisasi pada Gambar 4.2 adalah sebagai berikut:

1. Memasukkan data untuk melakukan perhitungan normalisasi.
2. Perulangan pada variabel i adalah untuk atribut dari penyakit gagal ginjal.
3. Perulangan pada variabel j adalah untuk semua data pada tiap atribut dari penyakit gagal ginjal.
4. Kondisi jika data pertama pada atribut pertama > max, bila kondisi tersebut benar maka disimpan pada variabel max dan bila tidak maka masuk kondisi berikutnya yaitu jika data pertama pada atribut pertama < min, bila kondisi tersebut benar maka disimpan pada variabel min dan bila tidak maka akan berulang hingga kondisi i dan j terpenuhi.
5. Perulangan pada variabel k adalah untuk perhitungan normalisasi data dari atribut pertama sebanyak data latih kemudian kembali ke perulangan i untuk atribut selanjutnya.

#### 4.2.3 Proses Algoritma Neighbor Weighted K-Nearest Neighbor (NWKNN)

Proses algoritma NWKNN yakni meliputi perhitungan nilai ketetanggaan dengan *euclidean distance*, pengurutan hasil ketetanggaan, pembobotan tiap jenis

identifikasi, perhitungan skor dan yang terakhir didapatkanlah hasil identifikasi gagal ginjal. Proses dari algoritma NWKNN dapat dilihat pada Gambar 4.3.



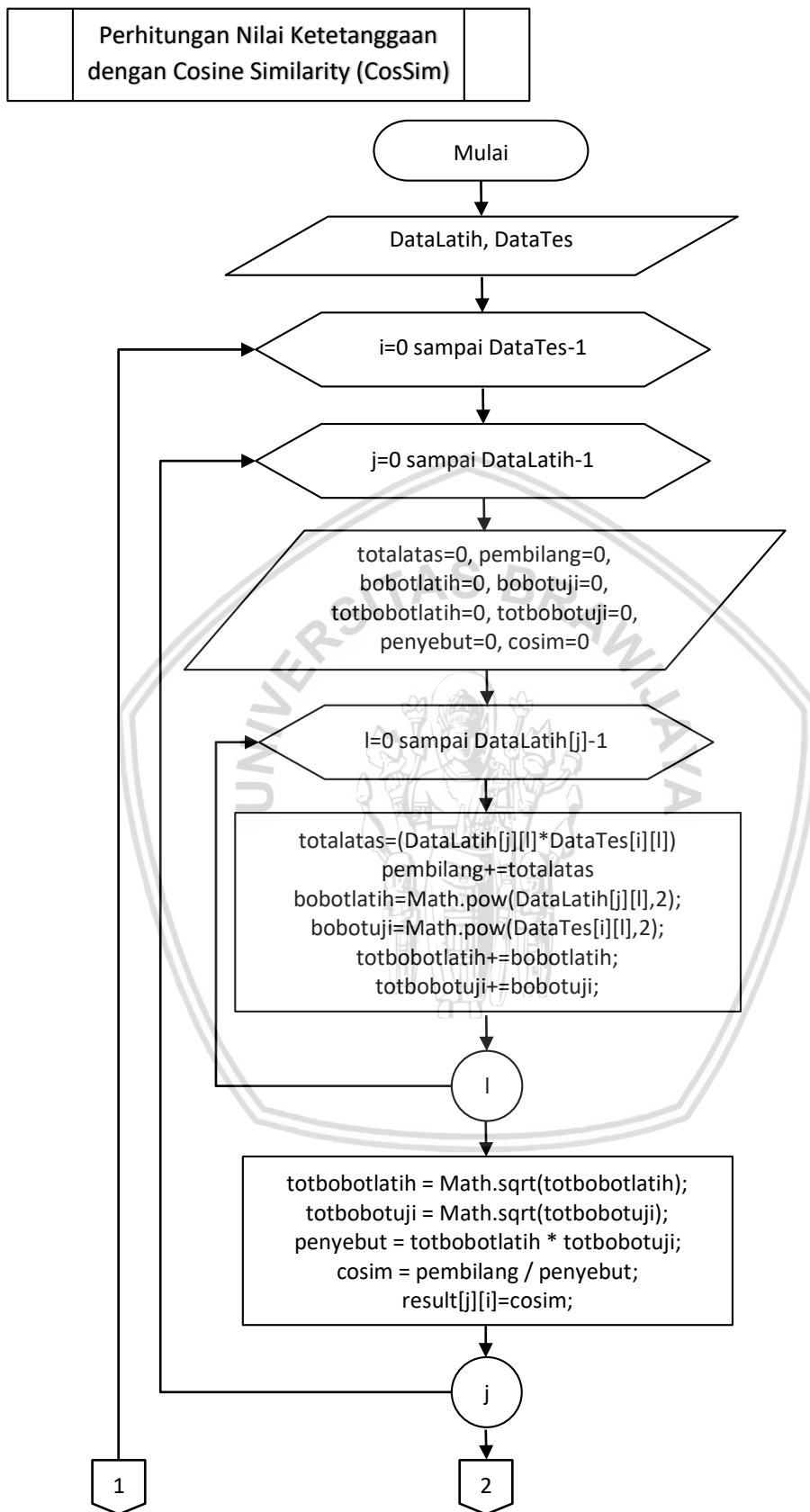
**Gambar 4.3 Proses NWKNN**

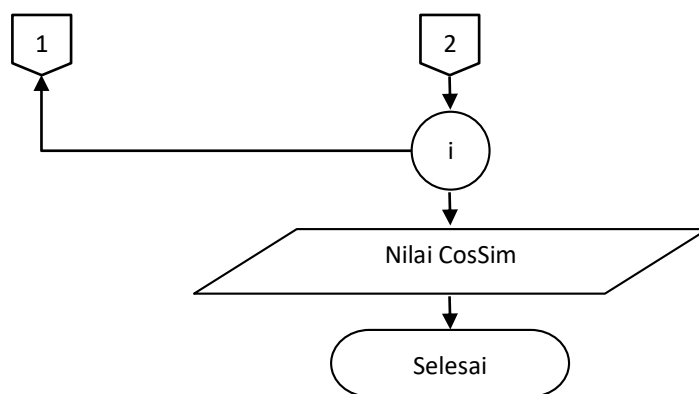
Langkah-langkah proses NWKNN pada Gambar 4.3 adalah sebagai berikut:

1. Masukkan data, nilai K dan nilai E.
2. Melakukan perhitungan nilai ketetanggaan menggunakan *Cosine Similarity (CosSim)* dan didapatkan hasil nilai kedekatan ketetanggaan.
3. Melakukan perhitungan pembobotan setiap kelas.
4. Melakukan perhitungan nilai skor.
5. Kemudian memperoleh hasil identifikasi gagal ginjal.

#### **4.2.4 Proses perhitungan ketetanggaan dengan *Cosine Similarity (CosSim)***

Proses perhitungan ketetanggaan dengan *cosine similarity (CosSim)* untuk menghitung jarak antara data latih dengan data uji. Langkah pertama yakni memasukkan data latih dan data uji. Selanjutnya mulai perhitungan data menggunakan *cosine similarity (CosSim)*. Dan yang terakhir didapatkan nilai kedekatan ketetanggaan antara data latih dan data uji. Proses dari perhitungan tersebut dapat dilihat pada Gambar 4.4.





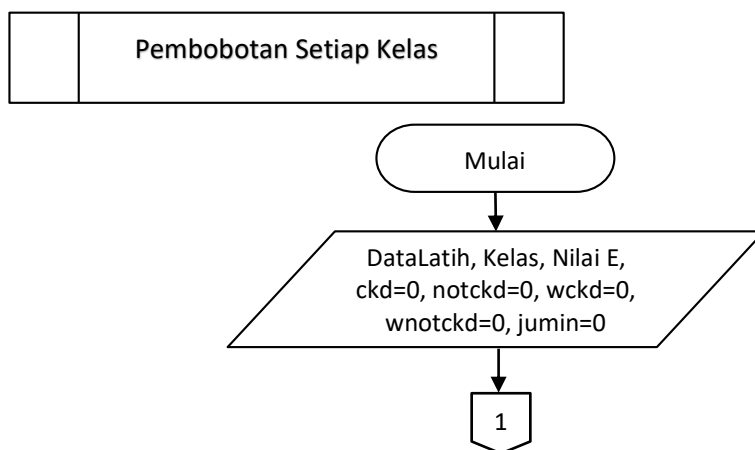
**Gambar 4.4** Proses perhitungan ketetangaan dengan *cosine similarity (CosSim)*

Langkah-langkah proses perhitungan ketetangaan pada Gambar 4.4 adalah sebagai berikut:

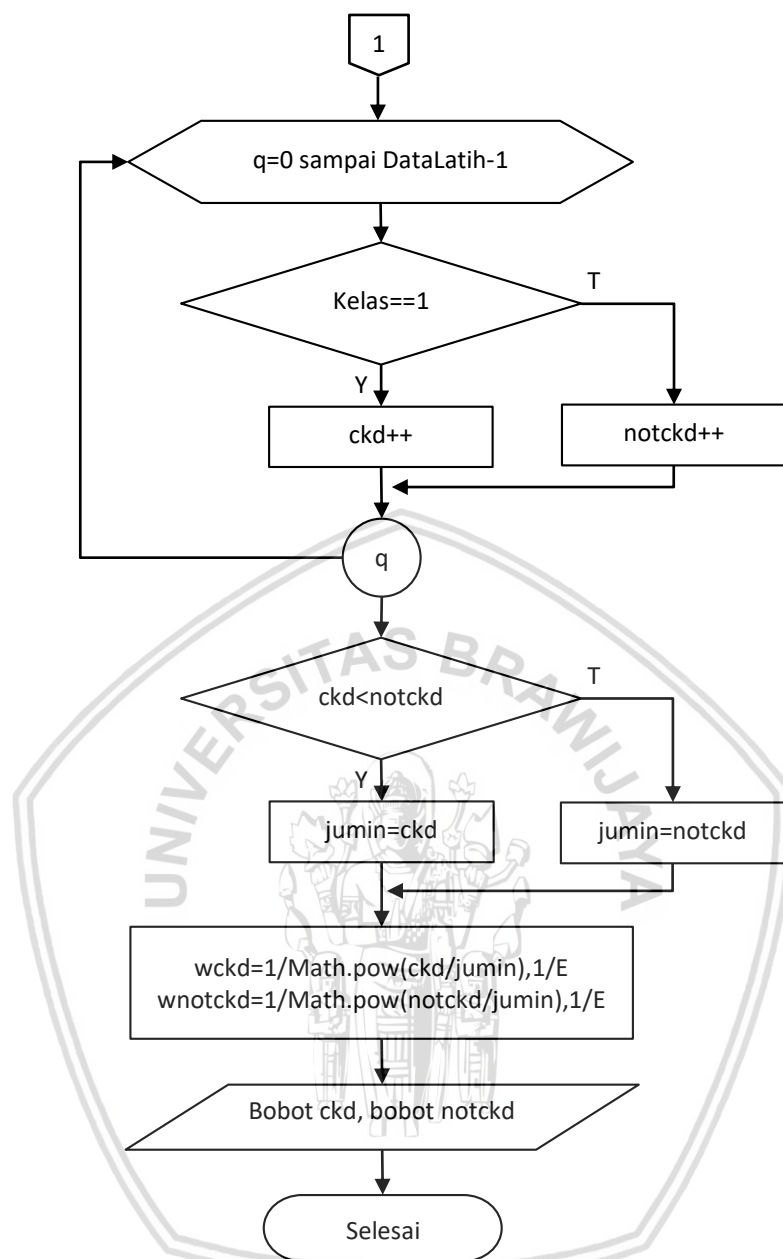
1. Inisialisasi data latih dan data tes.
2. Perulangan pada variabel  $j$  digunakan untuk semua data latih dan nilai variabel  $d$  serta temp akan bernilai 0 setelah perulangan ini dilakukan.
3. Perulangan pada variabel  $l$  adalah untuk perulangan atribut tiap data latih. Perhitungan kedekatan ketetangaan menggunakan *cossim* yakni yang pertama mengalikan antara nilai data latih pada semua atribut dengan nilai data uji pada semua atribut kemudian disimpan dalam variabel pembilang. Selanjutnya pada semua nilai data latih dan data uji dipangkat 2.
4. Kemudian hasil masing-masing dari data latih dan data uji tersebut di akar kuadrat. Selanjutnya hasil dari akar kuadrat dari masing-masing data latih dan data uji dikalikan dan disimpan pada variabel penyebut. Yang terakhir adalah variabel pembilang dibagi variabel penyebut dan dihasilkan nilai *cossim* yang disimpan pada variabel *cosim*. Hasil jarak tiap data disimpan pada array `result[j][1]`.

#### 4.2.5 Proses perhitungan bobot pada atribut *class*

Pada proses perhitungan bobot ini nantinya akan didapatkan hasil bobot dari setiap atribut *class*. Proses perhitungan bobot dapat dilihat pada Gambar 4.5.







**Gambar 4.5 Proses perhitungan bobot**

Langkah-langkah proses perhitungan bobot pada Gambar 4.5 adalah sebagai berikut:

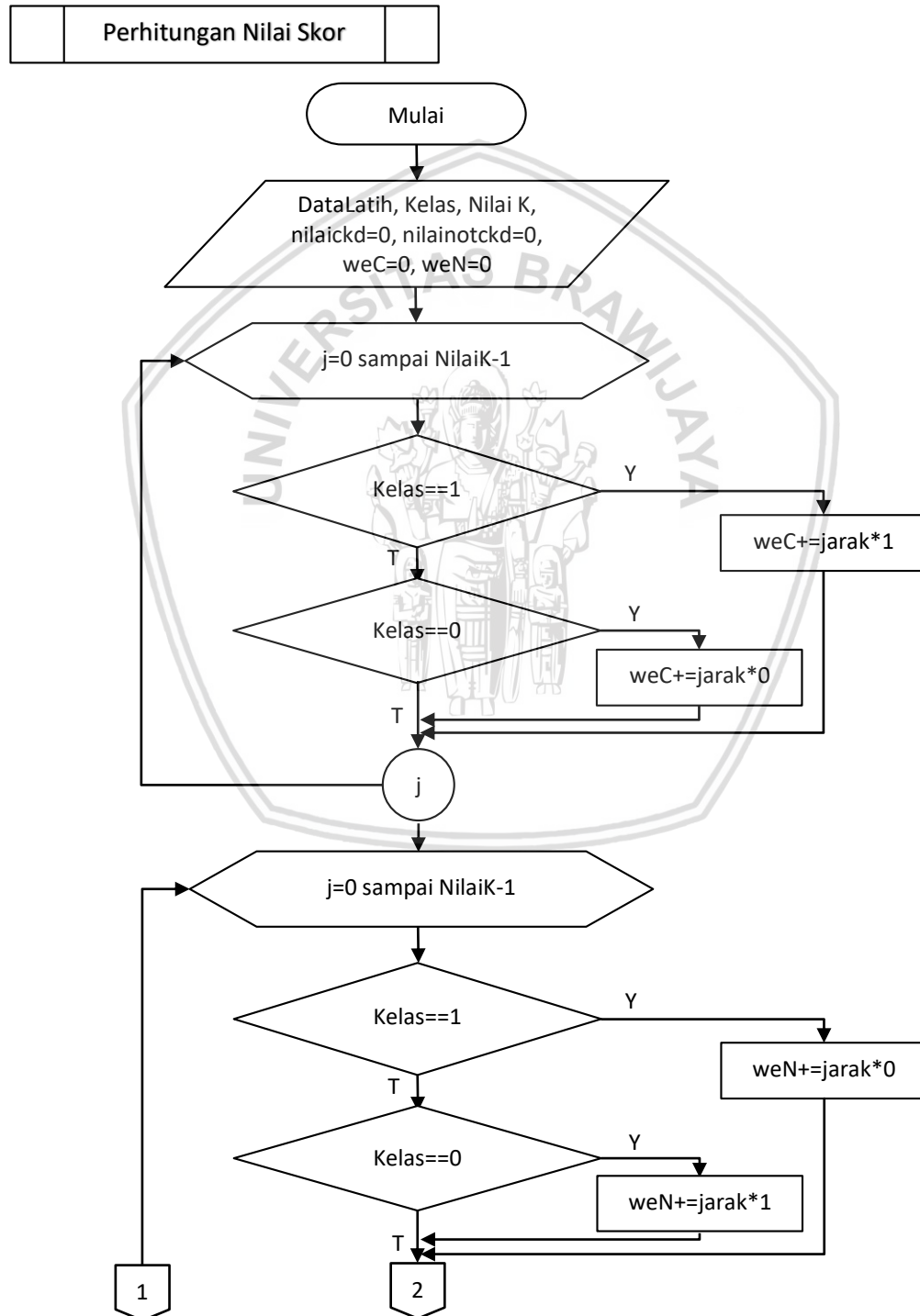
1. Memasukan data latih yang mana didalam data latih tersebut terdapat atribut kelas, nilai E sudah ditentukan dalam program, serta inisialisasi beberapa atribut.
2. Perulangan variabel q adalah untuk kelas pada tiap data latih.
3. Kondisi jika benar kelas = 1 maka bernilai ckd dan jika salah maka bernilai notckd. Kondisi ini berulang hingga akhir dari data latih.
4. Kondisi jika nilai  $ckd < notckd$  adalah benar maka nilai ckd dimasukkan variabel jumin, dan jika salah maka nilai notckd dimasukkan variabel jumin.

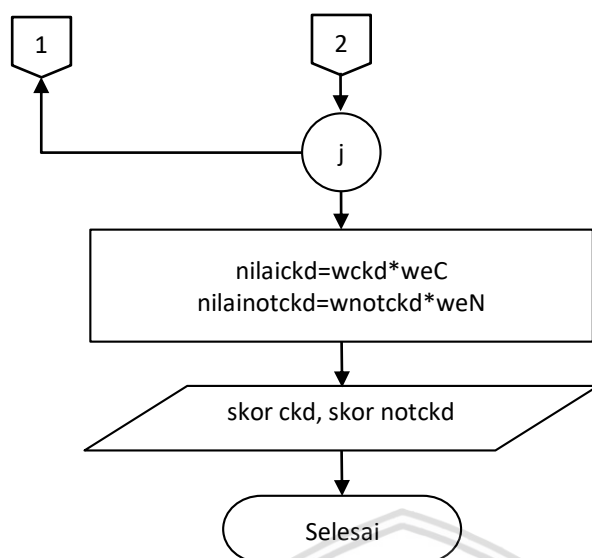
5. Selanjutnya menghitung nilai bobot tiap kelas dari ckd dan notckd yakni 1 dibagi jumlah kelas dari ckd atau notckd dibagi jumlah kelas yang paling sedikit kemudian dipangkat 1 dibagi nilai E.

6. Kemudian didapatkan nilai bobot tiap kelas dari ckd dan notckd.

#### 4.2.6 Proses perhitungan nilai skor

Setelah didapatkan nilai bobot, langkah berikutnya yakni perhitungan nilai skor. Proses perhitungan nilai skor dapat dilihat pada Gambar 4.6.





**Gambar 4.6 Proses perhitungan nilai skor**

Langkah-langkah proses perhitungan nilai skor pada Gambar 4.6 adalah sebagai berikut:

1. Memasukan data latih dan nilai K yang mana didalam data latih tersebut terdapat atribut kelas dan nilai K sudah ditentukan dalam program, serta inisialisasi beberapa atribut.
2. Melakukan pengecekan pada kelas ckd apakah kelas = 1, jika benar maka jarak yang sudah dihitung dengan *euclidean distance* akan dikali 1 dan berulang sebanyak nilai K serta disimpan pada variabel weC. Jika kelas = 0 maka jarak akan dikali 0 dan berulang sebanyak nilai k serta disimpan pada variabel weC.
3. Melakukan pengecekan pada kelas notckd apakah kelas = 1, jika benar maka jarak yang sudah dihitung dengan *euclidean distance* akan dikali 0 dan berulang sebanyak nilai K serta disimpan pada variabel weN. Jika kelas = 0 maka jarak akan dikali 1 dan berulang sebanyak nilai k serta disimpan pada variabel weN.
4. Melakukan perhitungan skor untuk kelas ckd dan notckd bobot dikali hasil perhitungan sebelumnya. Didapatkan hasil dari skor kelas ckd dan notckd.

### 4.3 Perhitungan manual

Perhitungan manual berfungsi untuk memberi gambaran umum perancangan alur proses perhitungan pada sistem yang akan dibuat. Data yang digunakan pada penelitian ini terdapat data berupa kategori maka diubah menjadi numerik supaya mudah untuk melakukan perhitungan. Pada atribut *red blood cells* dan *pus cell* yang bernilai normal diganti dengan 1 dan abnormal diganti dengan 0. Untuk atribut *pus cell clumps* dan *bacteria* yang bernilai *present* diganti 1 dan *notpresent* diganti 0. Pada atribut *appetite* bernilai *good* dan *poor* masing-masing diubah menjadi 1 dan 0. Pada atribut *hypertension*, *diabetes mellitus*, *coronary artery*

*disease, pedal edema, dan anemia* yang bernilai *yes* akan diganti 1 dan *no* akan diganti dengan 0. Untuk hasil identifikasi berupa *class* juga diubah menjadi 1 untuk ckd dan 0 untuk notckd.

Untuk mempermudah perhitungan maka nama atribut diringkas yang ditunjukkan pada Tabel 4.1.

**Tabel 4.1 Keterangan atribut**

No	Nama Atribut	Keterangan
1	Age	Age
2	Bp	Blood Pressure
3	Sg	Specific Gravity
4	Al	Albumin
5	Su	Sugar
6	Rbc	Red Blood Cells
7	Pc	Pus Cell
8	Pcc	Pus Cell Clumps
9	Ba	Bacteria
10	Bgr	Blood Glucose Random
11	Bu	Blood Urea
12	Sc	Serum Creatinine
13	Sod	Sodium
14	Pot	Potassium
15	Hemo	Hemoglobin
16	Pcv	Packed Cell Volume
17	Wbcc	White Blood Cell Count
18	Rbcc	Red Blood Cell Count
19	Htn	Hypertension
20	Dm	Diabetes Mellitus
21	Cad	Coronary Artery Disease
22	Appet	Appetite
23	Pe	Pedal Edema
24	Ane	Anemia

Perhitungan manual ini menggunakan sampel data dengan 20 data latih dan 5 data uji yang dapat dilihat pada Tabel 4.2.

Tabel 4.2 Sampel data latih dan data uji

No	Age	Bp	Sg	Al	Su	Rbc	Pc	Pcc	Ba	Bgr	Bu	Sc	Sod	Pot	Hemo	Pcv	Wbcc	Rbcc	Htn	Dm	Cad	Appet	Pe	Ane	Class
1	48	70	1.005	4	0	1	0	1	0	117	56	3.8	111	2.5	11.2	32	6700	3.9	1	0	0	0	1	1	1
2	53	90	1.02	2	0	0	0	1	0	70	107	7.2	114	3.7	9.5	29	12100	3.7	1	1	0	0	0	1	1
3	63	70	1.01	3	0	0	0	1	0	380	60	2.7	131	4.2	10.8	32	4500	3.8	1	1	0	0	1	0	1
4	68	80	1.01	3	2	1	0	1	1	157	90	4.1	130	6.4	5.6	16	11000	2.6	1	1	1	0	1	0	1
5	61	80	1.015	2	0	0	0	0	0	173	148	3.9	135	5.2	7.7	24	9200	3.2	1	1	1	0	1	1	1
6	48	80	1.025	4	0	1	0	0	0	95	163	7.7	136	3.8	9.8	32	6900	3.4	1	0	0	1	0	1	1
7	69	70	1.01	3	4	1	0	0	0	264	87	2.7	130	4	12.5	37	9600	4.1	1	1	1	1	1	0	1
8	73	70	1.005	0	0	1	1	0	0	70	32	0.9	125	4	10	29	18900	3.5	1	1	0	1	1	0	1
9	73	80	1.02	2	0	0	0	0	0	253	142	4.6	138	5.8	10.5	33	7200	4.3	1	1	1	1	0	0	1
10	59	60	1.02	0	0	1	1	0	0	113	23	1.1	139	3.5	15.3	54	6500	4.9	0	0	0	1	0	0	0
11	48	80	1.025	0	0	1	1	0	0	75	22	0.8	137	5	16.8	51	6000	6.5	0	0	0	1	0	0	0
12	80	80	1.025	0	0	1	1	0	0	119	46	0.7	141	4.9	13.9	49	5100	5	0	0	0	1	0	0	0
13	57	60	1.02	0	0	1	1	0	0	132	18	1.1	150	4.7	15.4	42	11000	4.5	0	0	0	1	0	0	0
14	63	70	1.02	0	0	1	1	0	0	113	25	0.6	146	4.9	16.5	52	8000	5.1	0	0	0	1	0	0	0
15	46	70	1.025	0	0	1	1	0	0	100	47	0.5	142	3.5	16.4	43	5700	6.5	0	0	0	1	0	0	0
16	15	80	1.025	0	0	1	1	0	0	93	17	0.9	136	3.9	16.7	50	6200	5.2	0	0	0	1	0	0	0
17	51	80	1.02	0	0	1	1	0	0	94	15	1.2	144	3.7	15.5	46	9500	6.4	0	0	0	1	0	0	0
18	41	80	1.025	0	0	1	1	0	0	112	48	0.7	140	5	17	52	7200	5.8	0	0	0	1	0	0	0
19	52	80	1.025	0	0	1	1	0	0	99	25	0.8	135	3.7	15	52	6300	5.3	0	0	0	1	0	0	0
20	36	80	1.025	0	0	1	1	0	0	85	16	1.1	142	4.1	15.6	44	5800	6.3	0	0	0	1	0	0	0
21	54	100	1.015	3	0	1	1	1	0	162	66	1.6	136	4.4	10.3	33	8260	4.2	1	1	0	0	1	0	1
22	53	90	1.015	0	0	1	1	0	0	168.6	38	2.2	139	4.24	10.9	34	4300	3.7	0	0	0	0	0	1	1
23	56	70	1.025	0	0	1	1	0	0	70	46	1.2	135	4.9	15.9	50	11000	5.1	0	0	0	1	0	0	0
24	54	70	1.02	0	0	1	1	0	0	76	28	0.6	146	3.5	14.8	52	8400	5.9	0	0	0	1	0	0	0
25	50	80	1.02	0	0	1	1	0	0	92	19	1.2	150	4.8	14.9	48	4700	5.4	0	0	0	1	0	0	0

### 4.3.1 Normalisasi data

Perhitungan normalisasi untuk menghitung sampel dari data latih dan data uji menggunakan normalisasi *min-max*. Berikut adalah contoh perhitungan normalisasi nilai terhadap atribut *age* yang terdapat pada Tabel 4.2.

$$\begin{aligned} V &= 48 \text{ (nilai pertama pada atribut } age) \\ \min_A &= 15 \text{ (nilai minimum pada atribut } age) \\ \max_A &= 80 \text{ (nilai maksimum pada atribut } age) \\ \max_A - \min_A &= 65 \text{ (hasil pengurangan } \max_A \text{ dan } \min_A) \end{aligned}$$

Kemudian dilakukan normalisasi *min-max* dengan persamaan 2.1:

$$\begin{aligned} V' &= \frac{V - \min_A}{\max_A - \min_A} \\ &= \frac{48 - 15}{65} \\ &= 0.508 \end{aligned}$$

Perhitungan diatas menghasilkan nilai normalisasi dari atribut *age* dengan nilai 0.508. Atribut yang lain juga mendapat perlakuan sama terhadap masing-masing nilai. Hasil dari seluruh hasil perhitungan normalisasi untuk data latih dapat dilihat pada Tabel 4.3.

Tabel 4.3 Normalisasi data

No	Age	Bp	Sg	Al	Su	...	Pe	Ane
1	0.507	0.25	0	1	0	...	1	1
2	0.584	0.75	0.75	0.5	0	...	0	1
3	0.738	0.25	0.25	0.75	0	...	1	0
4	0.815	0.5	0.25	0.75	0.5	...	1	0
5	0.707	0.5	0.5	0.5	0	...	1	1
6	0.507	0.5	1	1	0	...	0	1
7	0.830	0.25	0.25	0.75	1	...	1	0
8	0.892	0.25	0	0	0	...	1	0
9	0.892	0.5	0.75	0.5	0	...	0	0
10	0.676	0	0.75	0	0	...	0	0
11	0.507	0.5	1	0	0	...	0	0
12	1	0.5	1	0	0	...	0	0
13	0.646	0	0.75	0	0	...	0	0
14	0.738	0.25	0.75	0	0	...	0	0
15	0.476	0.25	1	0	0	...	0	0
16	0	0.5	1	0	0	...	0	0
17	0.553	0.5	0.75	0	0	...	0	0
18	0.4	0.5	1	0	0	...	0	0
19	0.569	0.5	1	0	0	...	0	0
20	0.323	0.5	1	0	0	...	0	0



No	Age	Bp	Sg	Al	Su	...	Pe	Ane
21	0.6	1	0.5	0.75	0	...	1	0
22	0.584	0.75	0.5	0	0	...	0	1
23	0.630	0.25	1	0	0	...	0	0
24	0.6	0.25	0.75	0	0	...	0	0
25	0.538	0.5	0.75	0	0	...	0	0

#### 4.3.2 Metode NWKNN

Setelah semua data sudah dinormalisasi, selanjutnya adalah perhitungan metode NWKNN. Berikut ini adalah contoh perhitungan manual metode NWKNN.

##### 1. Menentukan dataset

Dataset yang digunakan adalah data yang sudah dinormalisasi. Dataset yang digunakan pada manualisasi meliputi data latih dan data uji dengan ketentuan:

- Terdapat 24 atribut yang digunakan untuk identifikasi penyakit gagal ginjal.
- Terdapat 2 jenis identifikasi yakni *ckd* dan *notckd*.
- Terdapat 20 data latih (nomor 1-20) dan 5 data uji (nomor 21-25).

Dataset yang digunakan pada perhitungan manualisasi ini dapat dilihat pada Tabel 4.3.

##### 2. Menentukan nilai K tetangga

Pada manualisasi ini nilai K tetangga yang digunakan adalah K=5.

##### 3. Menghitung nilai kedekatan ketetanggaan

Untuk menghitung nilai kedekatan ketetanggaan menggunakan *cosine similarity (CosSim)*. Contoh yakni dari data uji nomor 21 ke data latih nomor 1:

$$d_j \bullet q =$$

$$\begin{aligned} & ((0.507 \cdot 0.6) + (0.25 \cdot 1) + (0 \cdot 0.5) + (1 \cdot 0.75) + (0 \cdot 0) + (1 \cdot 1) + (0 \cdot 1) + (1 \cdot 1) + (0 \cdot 0) + \\ & (0.151 \cdot 0.296) + (0.277 \cdot 0.344) + (0.458 \cdot 0.152) + (0 \cdot 0.641) + (0 \cdot 0.487) + \\ & (0.491 \cdot 0.412) + (0.421 \cdot 0.447) + (0.164 \cdot 0.271) + (0.333 \cdot 0.410) + (1 \cdot 1) + (0 \cdot 1) + \\ & (0 \cdot 0) + (0 \cdot 0) + (1 \cdot 1) + (1 \cdot 0)) \end{aligned}$$

$$d_j \bullet q = 6.087323$$

$$|d_j| \cdot |q| = \sqrt{((0.507^2) + (0.25^2) + (0^2) + (1^2) + (0^2) + (1^2) + (0^2) + (1^2) + (0^2) + (0.151^2) + (0.277^2) + (0.458^2) + (0^2) + (0^2) + (0.491^2) + (0.421^2) + (0.164^2) + (0.333^2) + (1^2) + (0^2) + (0^2) + (0^2) + (1^2) + (1^2))}$$

$$X$$

$$\sqrt{((0.6^2) + (1^2) + (0.5^2) + (0.75^2) + (0^2) + (1^2) + (1^2) + (1^2) + (0^2) + (0.296^2) + (0.344^2) + (0.152^2) + (0.641^2) + (0.487^2) + (0.412^2) + (0.447^2) + (0.271^2) + (0.410^2) + (1^2) + (1^2) + (0^2) + (0^2) + (1^2) + (0^2))}$$

$$|d_j| \cdot |q| = 8.333376$$

$$\text{CosSim}(q, d_j) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_{i=1}^m (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^m w_{ij}^2 \cdot \sum_{i=1}^m w_{iq}^2}} = \frac{6.087323}{8.333376} = 0.730475$$

Perhitungan yang sama juga dilakukan untuk data latih lain terhadap data uji. Hasil identifikasi dengan nilai 1 adalah teridentifikasi gagal ginjal (*ckd*) dan nilai 0 adalah tidak terkena gagal ginjal (*notckd*). Hasil perhitungan ketetanggaan terdekat dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Nilai kedekatan ketetanggaan**

No	Data Uji 21 ke data latih		Data Uji 22 ke data latih		Data Uji 23 ke data latih		Data Uji 24 ke data latih		Data Uji 25 ke data latih	
	Hasil CosSim	Identi fikasi	Hasil CosSim	Identi fikasi	Hasil CosSim	Identi fikasi	Hasil CosSim	Identi fikasi	Hasil CosSim	Identi fikasi
1	0.7304	1	0.5066	1	0.3532	1	0.3507	1	0.3383	1
2	0.6924	1	0.4879	1	0.3695	1	0.3100	1	0.3198	1
3	0.8158	1	0.3504	1	0.3294	1	0.3223	1	0.3449	1
4	0.7759	1	0.3939	1	0.3516	1	0.3035	1	0.341	1
5	0.6729	1	0.4915	1	0.3408	1	0.2968	1	0.3257	1
6	0.5891	1	0.6386	1	0.6269	1	0.5901	1	0.6063	1
7	0.7059	1	0.4351	1	0.5601	1	0.5514	1	0.5485	1
8	0.7658	1	0.5206	1	0.6737	1	0.6579	1	0.6328	1
9	0.6202	1	0.4410	1	0.5788	1	0.5379	1	0.5691	1
10	0.5606	0	0.7366	0	0.9718	0	0.9856	0	0.9638	0
11	0.5990	0	0.7599	0	0.9777	0	0.9747	0	0.9818	0
12	0.6241	0	0.7960	0	0.9714	0	0.9628	0	0.9749	0
13	0.5766	0	0.7478	0	0.9715	0	0.9688	0	0.9623	0
14	0.5994	0	0.7721	0	0.9837	0	0.9851	0	0.9859	0
15	0.5795	0	0.7409	0	0.9665	0	0.9834	0	0.9698	0
16	0.5749	0	0.7360	0	0.9533	0	0.9538	0	0.9600	0
17	0.6084	0	0.7661	0	0.9689	0	0.9905	0	0.9807	0
18	0.6095	0	0.7701	0	0.9833	0	0.9744	0	0.9826	0
19	0.6054	0	0.7714	0	0.9792	0	0.9819	0	0.9780	0
20	0.5966	0	0.7605	0	0.9649	0	0.9768	0	0.9824	0

#### 4. Mengurutkan hasil perhitungan kedekatan ketetanggan

Setelah didapatkan hasil kedekatan ketetanggan, langkah berikutnya yakni mengurutkan hasil perhitungan berdasar *cosine similarity (CosSim)* dari jarak terbesar ke terkecil. Hasil pengurutan dapat dilihat pada Tabel 4.5.

**Tabel 4.5 Urutan kedekatan ketetangga**

No	Data uji 21 ke data latih		Data uji 22 ke data latih		Data uji 23 ke data latih		Data uji 24 ke data latih		Data uji 25 ke data latih	
	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi
1	0.8158	1	0.7960	0	0.9837	0	0.9905	0	0.9859	0
2	0.7759	1	0.7721	0	0.9833	0	0.9856	0	0.9826	0
3	0.7658	1	0.7714	0	0.9792	0	0.9851	0	0.9824	0
4	0.7304	1	0.7701	0	0.9777	0	0.9834	0	0.9818	0
5	0.7059	1	0.7661	0	0.9718	0	0.9819	0	0.9807	0
6	0.6924	1	0.7605	0	0.9715	0	0.9768	0	0.9780	0
7	0.6729	1	0.7599	0	0.9714	0	0.9747	0	0.9749	0
8	0.6241	0	0.7478	0	0.9689	0	0.9744	0	0.9698	0
9	0.6202	1	0.7409	0	0.9665	0	0.9688	0	0.9638	0
10	0.6095	0	0.7366	0	0.9649	0	0.9628	0	0.9623	0
11	0.6084	0	0.7360	0	0.9533	0	0.9538	0	0.9600	0
12	0.6054	0	0.6386	1	0.6737	1	0.6579	1	0.6328	1
13	0.5994	0	0.5206	1	0.6269	1	0.5901	1	0.6063	1
14	0.5990	0	0.5066	1	0.5788	1	0.5514	1	0.5691	1
15	0.5966	0	0.4915	1	0.5601	1	0.5379	1	0.5485	1
16	0.5891	1	0.4879	1	0.3695	1	0.3507	1	0.3449	1
17	0.5795	0	0.4410	1	0.3532	1	0.3223	1	0.3416	1
18	0.5766	0	0.4351	1	0.3516	1	0.3100	1	0.3383	1
19	0.5749	0	0.3939	1	0.3408	1	0.3035	1	0.3257	1
20	0.5606	0	0.3504	1	0.3294	1	0.2968	1	0.3198	1

#### 5. Pembobotan setiap kelas identifikasi

Langkah selanjutnya pemberian bobot pada setiap kelas identifikasi (*ckd* dan *notckd*) dengan menggunakan nilai K dan nilai eksponen yang telah ditentukan. Pembobotan berdasar pada atribut kelas yang ada pada data latih. Jumlah nilai *ckd* dan *notckd* pada data latih yakni masing-masing sebesar 9 dan 11. Berikut merupakan contoh perhitungan dengan nilai K=5 dan eksponen=4 serta hasil pembobotan dapat dilihat pada Tabel 4.6.

$$Weight_i = \frac{1}{\left( \frac{\text{banyak data latih } i}{\text{jumlah data latih minimum pada } k} \right)^{\frac{1}{exp}}}$$

$$Weight_{(ckd)} = \frac{1}{\left( \frac{\text{jumlah data ckd}}{\text{jumlah data minimum}} \right)^{\frac{1}{exp}}} = \frac{1}{\left( \frac{9}{9} \right)^{\frac{1}{4}}} = 1$$

$$Weight_{(notckd)} = \frac{1}{\left(\frac{jumlah\ data\ notckd}{jumlah\ data\ minimum}\right)^{\frac{1}{exp}}} = \frac{1}{\left(\frac{11}{9}\right)^{\frac{1}{4}}} = 0.951$$

**Tabel 4.6 Nilai bobot setiap jenis identifikasi**

No	Jenis	Nilai Bobot
1	Ckd	1
2	Notckd	0.951

6. Perhitungan skor

Setelah mendapatkan nilai bobot, selanjutnya adalah melakukan perhitungan skor. Hasil skor yang paling besar akan menjadi hasil identifikasi. Sebelum perhitungan skor, data yang sudah diurutkan berdasar jarak diambil dan dipilih berdasar nilai K. Berikut ini merupakan contoh perhitungan skor.

- a. Mengambil data yang sudah diurutkan berdasar jarak dengan nilai K=5

**Tabel 4.7 Jarak yang telah diurutkan dengan nilai K=5**

No	Data uji 21 ke data latih		Data uji 22 ke data latih		Data uji 23 ke data latih		Data uji 24 ke data latih		Data uji 25 ke data latih	
	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi	Hasil Jarak Urut	Identi fikasi
1	0.8158	1	0.7960	0	0.9837	0	0.9905	0	0.9859	0
2	0.7759	1	0.7721	0	0.9833	0	0.9856	0	0.9826	0
3	0.7658	1	0.7714	0	0.9792	0	0.9851	0	0.9824	0
4	0.7304	1	0.7701	0	0.9777	0	0.9834	0	0.9818	0
5	0.7059	1	0.7661	0	0.9718	0	0.9819	0	0.9807	0

- b. Melakukan perhitungan skor pada setiap kelas Identifikasi yang masuk dalam K tetangga.

Contoh perhitungan skor dilakukan pada data uji ke 21. Pada hasil jarak yang sudah diurutkan pada proses sebelumnya, maka pada data ke 21 terdapat identifikasi ckd dan notckd, maka yang akan akan dihitung adalah kedua identifikasi tersebut.

$$Skor(X, C_i) = Weight_i *$$

$$\left(\sum_{d_j \in NWKNN(X)} \left( \left( \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2} \right) * \delta(d_j, C_i) \right) \right)$$

$$Skor(ckd) =$$

$$1 * ((0.8158 * 1) + (0.7759 * 1) + (0.7658 * 1) + (0.7304 * 1) + (0.7059 * 1)) = 3.7939$$

$$Skor(notckd) =$$

$$0.951 * ((0.8158 * 0) + (0.7759 * 0) + (0.7658 * 0) + (0.7304 * 0) + (0.7059 * 0)) = 0$$

Berdasarkan perhitungan skor, data uji ke 21 termasuk dalam identifikasi ckd/terkena gagal ginjal karena nilai terbesar mengarah pada skor ckd.

Hasil dari perhitungan skor data uji ke 21 sampai data uji ke 25 dapat dilihat pada Tabel 4.8.

**Tabel 4.8 Hasil perhitungan skor**

No	Data uji 21 ke data latih				
	Jenis	Skor	Data Asli	Hasil Identifikasi	Kecocokan
1	<u>ckd</u>	<u>3.7939</u>	ckd	ckd	valid
2	notckd	0			
No	Data uji 22 ke data latih				
	Jenis	Skor	Data Asli	Hasil Identifikasi	Kecocokan
1	ckd	0	ckd	notckd	tidak valid
2	<u>notckd</u>	<u>3.6863</u>			
No	Data uji 23 ke data latih				
	Jenis	Skor	Data Asli	Hasil Identifikasi	Kecocokan
1	ckd	0	notckd	notckd	valid
2	<u>notckd</u>	<u>4.6564</u>			
No	Data uji 24 ke data latih				
	Jenis	Skor	Data Asli	Hasil Identifikasi	Kecocokan
1	ckd	0	notckd	notckd	valid
2	<u>notckd</u>	<u>4.6857</u>			
No	Data uji 25 ke data latih				
	Jenis	Skor	Data Asli	Hasil Identifikasi	Kecocokan
1	ckd	0	notckd	notckd	valid
2	<u>notckd</u>	<u>4.6732</u>			

Berdasarkan pada perhitungan manualisasi, maka hasil kecocokan data asli dengan hasil identifikasi dari sistem ada 4 dari 5 hasil uji coba. Maka dapat diketahui nilai akurasi dari sistem, berikut merupakan contoh perhitungan akurasi:

$$akurasi = \frac{\text{jumlah identifikasi benar}}{\text{jumlah data}} \times 100\%$$

$$akurasi = \frac{4}{5} \times 100\%$$

$$akurasi = 80\%$$

#### 4.4 Perancangan pengujian

Pada sub bab ini akan menjelaskan beberapa perancangan pengujian yakni pengujian pengaruh nilai K terhadap akurasi, pengujian pengaruh nilai E terhadap akurasi, dan perancangan pengujian perbandingan akurasi metode KNN dengan NWKNN.

##### 4.4.1 Perancangan pengujian pengaruh nilai K dan nilai E

Pengujian pengaruh nilai K dan nilai E dilakukan untuk mengetahui bagaimana akurasi sistem jika nilai K dan nilai E berubah-ubah. Akan terdapat 4 kali skenario pada pengujian ini, skenario pertama yakni nilai K diubah dari 2 hingga 28, nilai E=2 dan menggunakan semua komposisi data latih yakni 150, 125 dan 100 data. Kemudian untuk skenario kedua, ketiga dan keempat yang berbeda adalah nilai E, skenario kedua menggunakan nilai E=4, skenario ketiga nilai E=6 dan skenario keempat nilai E=8. Perancangan pengujian pengaruh nilai K dan nilai E terhadap akurasi dapat dilihat pada Tabel 4.9.

**Tabel 4.9 Skenario perancangan pengujian pengaruh nilai K dan nilai E**

Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
2			
3			
4			
5			
6			
7			
9			
11			
13			
15			
17			
19			
20			
21			
23			
25			
27			
28			



#### 4.4.2 Perancangan pengujian perbandingan akurasi metode NWKNN berdasarkan pengukuran jarak

Pengujian ini bertujuan mengetahui hasil akurasi identifikasi penyakit gagal ginjal dengan metode NWKNN berdasarkan pengukuran jarak dengan membandingkan nilai *Euclidean Distance* dan *CosSim*. Perancangan pengujian perbandingan akurasi metode NWKNN berdasarkan pengukuran jarak dapat dilihat pada Tabel 4.10.

**Tabel 4.10 Perancangan pengujian perbandingan akurasi metode NWKNN berdasarkan pengukuran jarak**

Nilai K	Tingkat Akurasi <i>Euclidean Distance</i> (%)	Tingkat Akurasi <i>CosSim</i> (%)
2		
3		
4		
5		
6		
7		
9		
11		
13		
15		
17		
19		
20		
21		
23		
25		
27		
28		
Hasil Rata-Rata Akurasi		

#### 4.4.3 Perancangan pengujian perbandingan akurasi metode NWKNN dan KNN

Pengujian ini bertujuan mengetahui hasil akurasi identifikasi penyakit gagal ginjal dengan metode NWKNN dan dibandingkan dengan metode KNN. Perancangan

pengujian perbandingan akurasi metode KNN dan NWKNN dapat dilihat pada Tabel 4.11.

**Tabel 4.11 Perancangan pengujian perbandingan akurasi metode NWKNN dan KNN**

Nilai K	Tingkat Akurasi NWKNN	Tingkat Akurasi KNN
2		
3		
4		
5		
6		
7		
9		
11		
13		
15		
17		
19		
20		
21		
23		
25		
27		
28		
Hasil Rata-Rata Akurasi		

## BAB 5 IMPLEMENTASI

Pada bab ini membahas mengenai implementasi sistem yang menerapkan rancangan yang telah dibuat pada bab sebelumnya yakni bab perancangan. Pembahasan mengenai implementasi meliputi implementasi program.

### 5.1 Implementasi program

Pada sub bab ini akan menjelaskan tentang kode program dari identifikasi penyakit gagal ginjal menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN).

#### 5.1.1 Implementasi proses normalisasi data

Potongan program proses normalisasi data ditunjukkan pada *Source Code 5.1*.

```

1  public void normalisasi(double[][] data) {
2      this.datanorm =
3      new double[data.length][data[0].length];
4      this.hasildatanorm =
5      new double[data.length][data[0].length];
6      this.datanorm = data;
7      this.low = new double[24];
8      this.mak = new double[24];
9      double[][] resul = new double[150][24];
10
11     for (int i = 0; i < datanorm[0].length; i++) {
12         double max = 0;
13         double min = 10000;
14         for (int j = 0; j < datanorm.length; j++) {
15             if (datanorm[j][i] > max) {
16                 max = datanorm[j][i];
17                 mak[i] = max;
18             }
19             if (datanorm[j][i] < min) {
20                 min = datanorm[j][i];
21                 low[i] = min;
22             }
23         }
24         for (int k = 0; k < datanorm.length; k++) {
25             hasildatanorm[k][i] = (datanorm[k][i] - min)
26             / (max - min);
27         }
28     }
29 }
```

**Source Code 5.1 Proses Normalisasi Data**

*Source Code 5.1* menjelaskan proses dari normalisasi data. Penjelasan dari *Source Code 5.1* adalah sebagai berikut:

1. Baris 1, deklarasi method normalisasi.
2. Baris 2-9, deklarasi variabel yang dibutuhkan pada proses normalisasi.
3. Baris 11, perulangan untuk pengecekan atribut dari penyakit gagal ginjal.
4. Baris 12-13, deklarasi variabel max dan min.

5. Baris 14, perulangan untuk pengecekan semua data pada tiap atribut dari penyakit gagal ginjal.
6. Baris 15-22, kondisi jika data pertama pada atribut pertama > max, bila kondisi tersebut benar maka disimpan pada variabel max dan bila tidak maka masuk kondisi berikutnya yaitu jika data pertama pada atribut pertama < min, bila kondisi tersebut benar maka disimpan pada variabel min dan bila tidak maka akan berulang hingga kondisi i dan j terpenuhi.
7. Baris 24-26, perulangan untuk perhitungan normalisasi data dari atribut pertama sebanyak data latih kemudian kembali ke perulangan I untuk atribut selanjutnya. Kemudian rumus atau perhitungan nilai normalisasi data disimpan pada variabel hasildataanorm.

### 5.1.2 Implementasi proses perhitungan kedekatan ketetangaan

Potongan program proses perhitungan kedekatan ketetangaan menggunakan *cosine similarity* ditunjukkan pada *Source Code 5.2*.

```

1  public void cossim(double datat[][], double kelast[]) {
2      this.hasil_testing = new double[kelast.length];
3      this.data_tes =
4      new double[datat.length][datat[0].length];
5      this.kelas_tes = new double[kelast.length];
6      this.jarak =
7      new double[data_latih.length][datat.length];
8      this.jarakurut =
9      new double[data_latih.length][datat.length];
10     this.skor = new double[datat.length][2];
11     this.data_tes = datat;
12     this.kelas_tes = kelast;
13     double[][] result =
14     new double[data_latih.length][3];
15
16     for (int i = 0; i < data_tes.length; i++) {
17         for (int j = 0; j < data_latih.length; j++) {
18             double totalatas=0, pembilang=0,
19             bobotlatih=0, bobotuji=0, totbobotlatih=0,
20             totbobotuji=0, penyebut=0, cosim=0;
21             for(int l=0; l<data_latih[j].length; l++) {
22                 totalatas = (data_latih[j][l] *
23                 data_tes[i][l]);
24                 pembilang += totalatas;
25                 bobotlatih =
26                 Math.pow(data_latih[j][l], 2);
27                 bobotuji = Math.pow(data_tes[i][l], 2);
28                 totbobotlatih += bobotlatih;
29                 totbobotuji += bobotuji;
30             }
31             totbobotlatih = Math.sqrt(totbobotlatih);
32             totbobotuji = Math.sqrt(totbobotuji);
33             penyebut = totbobotlatih * totbobotuji;
34             cosim = pembilang / penyebut;
35             result[j][1] = cosim;
36         }}}

```

**Source Code 5.2 Proses Perhitungan Kedekatan Ketetangaan**

Source Code 5.2 menjelaskan proses dari metode NWKNN yakni menghitung kedekatan ketetanggaan menggunakan *cosine similarity*. Penjelasan Source Code 5.2 adalah sebagai berikut:

1. Baris 1, deklarasi method *cossim*.
2. Baris 2-14, deklarasi variabel yang dibutuhkan pada proses perhitungan kedekatan ketetanggaan.
3. Baris 17, perulangan untuk pengecekan semua data latih.
4. Baris 18-20, deklarasi nilai variabel yang digunakan pada proses perhitungan kedekatan ketetanggaan.
5. Baris 21, perulangan untuk pengecekan pada atribut tiap data latih.
6. Baris 22-24, perhitungan kedekatan ketetanggaan menggunakan *cossim* yakni yang pertama mengalikan antara nilai data latih pada semua atribut dengan nilai data uji pada semua atribut kemudian disimpan dalam variabel pembilang.
7. Baris 25-33, selanjutnya pada semua nilai data latih dan data uji dipangkat 2 kemudian hasil masing-masing dari data latih dan data uji tersebut di akar kuadrat. Selanjutnya hasil dari akar kuadrat dari masing-masing data latih dan data uji dikalikan dan disimpan pada variabel penyebut.
8. Baris 34-35, yang terakhir adalah variabel pembilang dibagi variabel penyebut dan dihasilkan nilai *cossim* yang disimpan pada array *result*.

### 5.1.3 Implementasi proses perhitungan pembobotan kelas

Potongan program proses perhitungan pembobotan setiap kelas identifikasi ditunjukkan pada Source Code 5.3.

```

1  double ckd = 0;
2  double notckd = 0;
3  double wckd = 0;
4  double wnotckd = 0;
5  double jumin = 0;
6  for (int q = 0; q < result.length; q++) {
7      if (result[q][2] == 1) {
8          ckd++;
9      } else {
10         notckd++;
11     }
12 }
13     if (ckd < notckd) {
14         jumin = ckd;
15     } else {
16         jumin = notckd;
17     }
18     wckd = 1 / Math.pow(ckd / jumin, 1 / e);
19     wnotckd = 1 / Math.pow(notckd / jumin, 1 / e);

```

**Source Code 5.3 Proses Perhitungan Pembobotan Kelas**

*Source Code* 5.3 menjelaskan proses dari metode NWKNN selanjutnya yakni menghitung pembobotan kelas. Penjelasan *Source Code* 5.3 adalah sebagai berikut:

1. Baris 1-5, inialisasi variabel yang digunakan untuk proses pembobotan kelas.
2. Baris 6, perulangan untuk pengecekan kelas pada tiap data latih.
3. Baris 7-10, kondisi jika benar kelas = 1 maka bernilai ckd dan jika salah maka bernilai notckd. Kondisi ini berulang hingga akhir dari data latih.
4. Baris 13-16, kondisi jika nilai ckd kurang dari notckd adalah benar maka nilai ckd dimasukkan variabel jumin, dan jika salah maka nilai notckd dimasukkan variabel jumin.
5. Baris 18-19, menghitung nilai bobot tiap kelas dari ckd dan notckd yakni 1 dibagi jumlah kelas dari ckd atau notckd dibagi jumlah kelas yang paling sedikit kemudian dipangkat 1 dibagi nilai e. Selanjutnya masing-masing disimpan dalam variabel wckd dan wnotckd.

#### 5.1.4 Implementasi proses perhitungan skor

Potongan program proses perhitungan pembobotan setiap kelas identifikasi ditunjukkan pada *Source Code* 5.4.

```

1  double nilaickd = 0;
2  double nilainotckd = 0;
3  double weC = 0;
4  double weN = 0;
5
6  for (int j = 0; j < k; j++) {
7      if (result[j][2] == 1) {
8          weC += result[j][1] * 1;
9      } else if (result[j][2] == 0) {
10         weC += result[j][1] * 0;
11     }
12 }
13 for (int j = 0; j < k; j++) {
14     if (result[j][2] == 1) {
15         weN += result[j][1] * 0;
16     } else if (result[j][2] == 0) {
17         weN += result[j][1] * 1;
18     }
19 }
20 nilaickd = wckd * weC;
21 nilainotckd = wnotckd * weN;

```

**Source Code 5.4 Proses Perhitungan Pembobotan Kelas**

*Source Code* 5.4 menjelaskan tentang perhitungan skor. Penjelasan *Source Code* 5.4 adalah sebagai berikut:

1. Baris 1-4, inialisasi variabel yang dibutuhkan pada proses perhitungan skor.



2. Baris 6-10, melakukan pengecekan pada kelas ckd apakah kelas = 1, jika benar maka jarak yang sudah dihitung dengan *euclidean distance* akan dikali 1 dan berulang sebanyak nilai k serta disimpan pada variabel weC. Jika kelas = 0 maka jarak akan dikali 0 dan berulang sebanyak nilai k serta disimpan pada variabel weC.
3. Baris 13-17, melakukan pengecekan pada kelas notckd apakah kelas = 1, jika benar maka jarak yang sudah dihitung dengan *euclidean distance* akan dikali 0 dan berulang sebanyak nilai k serta disimpan pada variabel weN. Jika kelas = 0 maka jarak akan dikali 1 dan berulang sebanyak nilai k serta disimpan pada variabel weN.
4. Baris 20-21, melakukan perhitungan skor untuk kelas ckd dan notckd bobot dikali hasil perhitungan sebelumnya. Didapatkan hasil dari skor kelas ckd dan notckd.



## BAB 6 PENGUJIAN

Pada bab pengujian ini menjelaskan hasil dari proses pengujian terhadap sistem yang telah dibuat dengan menggunakan metode *neighbor weighted k-nearest neighbor* (NWKNN). Proses pengujian dilakukan berdasarkan perancangan yang telah dibuat sebelumnya. Proses pengujian meliputi pengaruh perubahan nilai K dan nilai E, perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak, dan perbandingan akurasi antara metode NWKNN dan KNN. Pengujian ini dilakukan untuk mengetahui hasil akurasi dari implementasi yang telah dilakukan.

### 6.1 Pengujian nilai K dan nilai E

Pengujian pengaruh nilai K dan nilai E ini dilakukan untuk mengetahui seberapa besar pengaruh nilai K dan nilai E yang bervariasi atau dengan nilai yang berubah terhadap akurasi sistem. Pada pengujian ini dilakukan dengan mengubah nilai K mulai dari 2 hingga 28, nilai E (eksponen) yang digunakan yakni 2, 4, 6 dan 8 serta data uji yang digunakan sebanyak 50 data. Pada penelitian ini terdapat 2 macam kelas identifikasi yakni CKD atau terkena gagal ginjal dan NOTCKD atau tidak terkena gagal ginjal yang mana jumlah datanya tidak seimbang atau terdapat kelas mayoritas dan minoritas. Pada pengujian ini komposisi data latih yang digunakan sebanyak 3 kali dengan terdapat kelas mayoritas dan minoritas untuk mendapatkan data mana yang memiliki akurasi tertinggi. Komposisi data latih yang digunakan adalah sebagai berikut:

- 150 data latih (CKD sebanyak 43 dan NOTCKD sebanyak 107),
- 125 data latih (CKD sebanyak 33 dan NOTCKD sebanyak 92),
- 100 data latih (CKD sebanyak 23 dan kelas NOTCKD sebanyak 77)

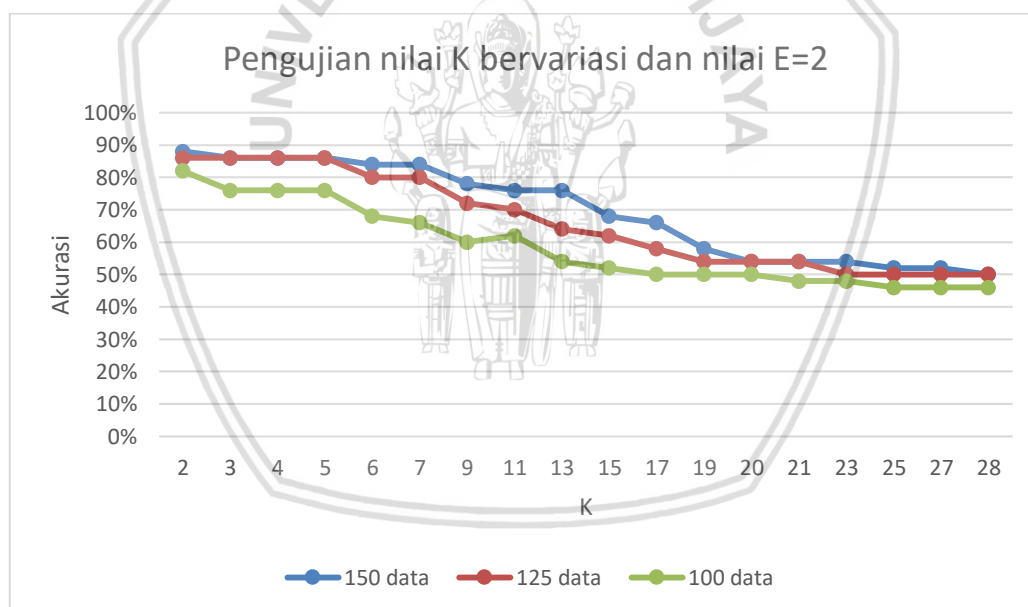
Pada pengujian nilai K dan nilai E ini dilakukan 4 kali skenario dengan skenario pertama yakni nilai K diubah dari 2 hingga 28, nilai E=2 dan menggunakan semua komposisi data latih yakni 150, 125 dan 100 data. Kemudian untuk skenario kedua, ketiga dan keempat yang berbeda adalah nilai E, skenario kedua menggunakan nilai E=4, skenario ketiga nilai E=6 dan skenario keempat nilai E=8.

Pengujian dengan skenario pertama yakni dilakukan dengan nilai E=2 dan nilai K berubah-ubah. Hasil nilai pengujian pertama dapat dilihat pada Tabel 6.1.

**Tabel 6.1 Hasil pengujian skenario 1 nilai E = 2**

Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
2	88%	86%	82%
3	86%	86%	76%
4	86%	86%	76%
5	86%	86%	76%

Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
6	84%	80%	68%
7	84%	80%	66%
9	78%	72%	60%
11	76%	70%	62%
13	76%	64%	54%
15	68%	62%	52%
17	66%	57.9%	50%
19	57.9%	54%	50%
20	54%	54%	50%
21	54%	54%	48%
23	54%	50%	48%
25	52%	50%	46%
27	52%	50%	46%
28	50%	50%	46%



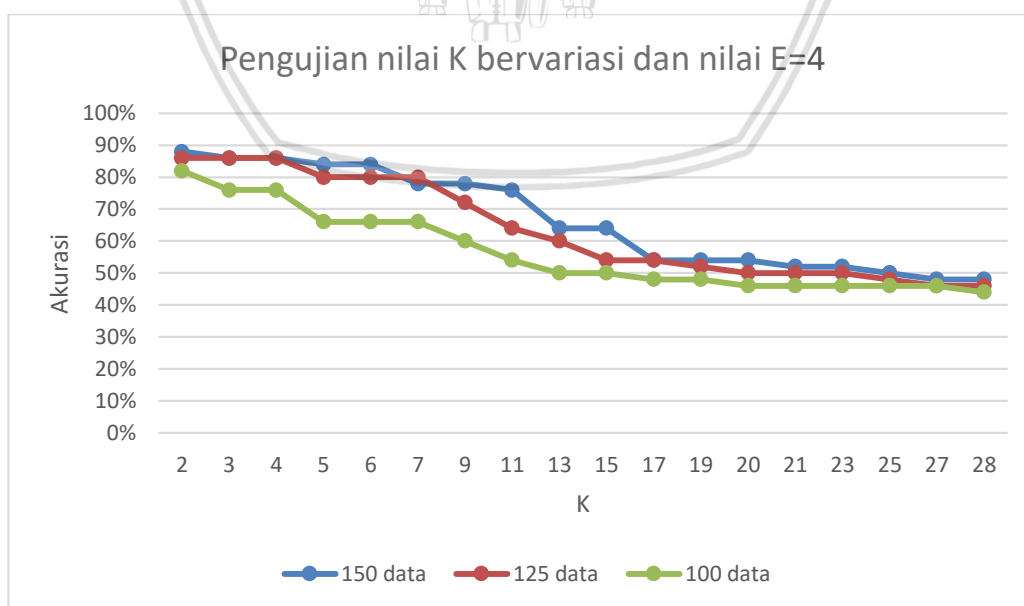
**Gambar 6.1** Grafik pengujian nilai K bervariasi dan nilai E=2

Berdasar pada Tabel 6.1 dan Gambar 6.1, nilai akurasi sistem tertinggi adalah pada nilai K=2, nilai E=2 dan data latih 150 yakni sebesar 88%. Pada pengujian skenario pertama ini semakin besar nilai K pada semua komposisi data latih maka nilai akurasi semakin menurun. Komposisi data latih dengan 150 data memiliki nilai akurasi tertinggi dibandingkan komposisi data latih 125 dan 100 dikarenakan semakin banyak data latih yang digunakan, maka semakin banyak data yang akan dibandingkan dengan data uji dan memungkinkan untuk memperoleh hasil CosSim yang mempunyai similarity terbaik.

Pengujian dengan skenario kedua yakni dilakukan dengan nilai  $E = 4$  dan nilai  $K$  berubah-ubah. Hasil nilai pengujian pertama dapat dilihat pada Tabel 6.2.

**Tabel 6.2 Hasil pengujian skenario 2 nilai  $E = 4$**

Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
2	88%	86%	82%
3	86%	86%	76%
4	86%	86%	76%
5	84%	80%	66%
6	84%	80%	66%
7	78%	80%	66%
9	78%	72%	60%
11	76%	64%	54%
13	64%	60%	50%
15	64%	54%	50%
17	54%	54%	48%
19	54%	52%	48%
20	54%	50%	46%
21	52%	50%	46%
23	52%	50%	46%
25	50%	48%	46%
27	48%	46%	46%
28	48%	46%	44%



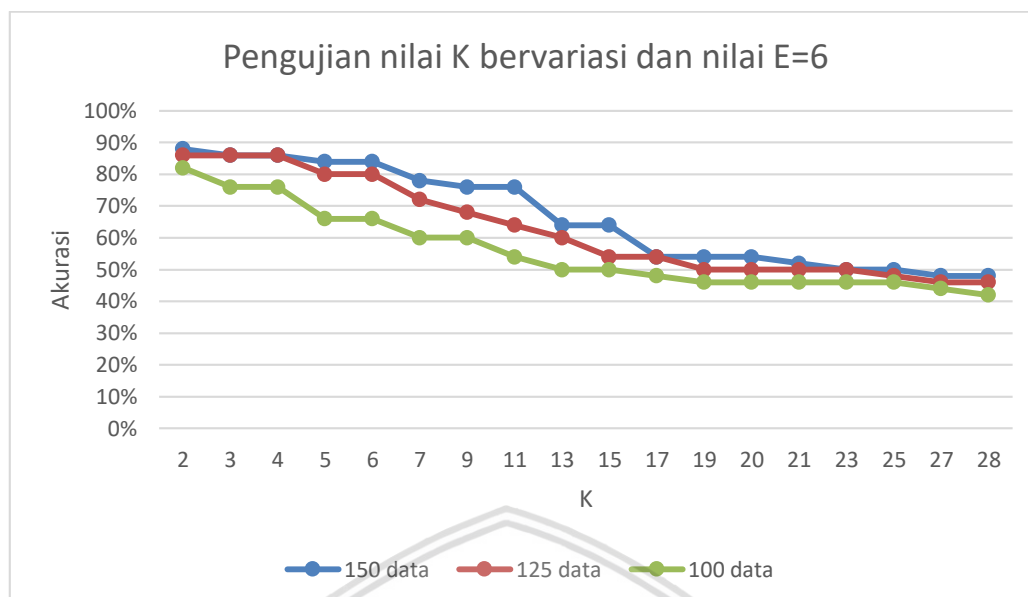
**Gambar 6.2 Grafik pengujian nilai K bervariasi dan nilai  $E=4$**

Berdasarkan pada Tabel 6.2 dan Gambar 6.2, skenario kedua dengan nilai  $E=4$  memiliki akurasi tertinggi yaitu 88% pada data latih sebanyak 150 dengan nilai  $K=2$ . Sama dengan skenario sebelumnya yakni semakin besar nilai  $K$ , maka semakin kecil nilai akurasi. Hal ini disebabkan identifikasi kelas memiliki perbedaan dengan kelas awal. Contohnya pada data uji pertama pada kelas awal yakni masuk kelas CKD atau terkena gagal ginjal, sedangkan hasil dari identifikasi sistem adalah NOTCKD atau tidak terkena gagal ginjal. Kelas yang berbeda ini terus bertambah seiring bertambahnya nilai  $K$  yang menjadikan tingkat akurasi semakin menurun. Kemudian komposisi data latih sebanyak 150 data masih memiliki akurasi tertinggi dibandingkan komposisi data latih yang lainnya.

Pengujian dengan skenario ketiga yakni dilakukan dengan nilai  $E=6$  dan nilai  $K$  berubah-ubah. Hasil nilai pengujian pertama dapat dilihat pada Tabel 6.3.

**Tabel 6.3 Hasil pengujian skenario 3 nilai  $E=6$**

Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
2	88%	86%	82%
3	86%	86%	76%
4	86%	86%	76%
5	84%	80%	66%
6	84%	80%	66%
7	78%	72%	60%
9	76%	68%	60%
11	76%	64%	54%
13	64%	60%	50%
15	64%	54%	50%
17	54%	54%	48%
19	54%	50%	46%
20	54%	50%	46%
21	52%	50%	46%
23	50%	50%	46%
25	50%	48%	46%
27	48%	46%	44%
28	48%	46%	42%



**Gambar 6.3 Grafik pengujian nilai K bervariasi dan nilai E=6**

Berdasar pada Tabel 6.3 dan Gambar 6.3, dengan nilai E=6 nilai akurasi tertinggi adalah 88% pada data latih 150 dengan nilai K=2. Grafik pengujian masih menunjukkan bahwa semakin besar nilai K maka semakin turun pula nilai akurasi, dan juga nilai akurasi pada komposisi data latih sebanyak 150 masih unggul dibandingkan komposisi data latih 125 dan 100 data.

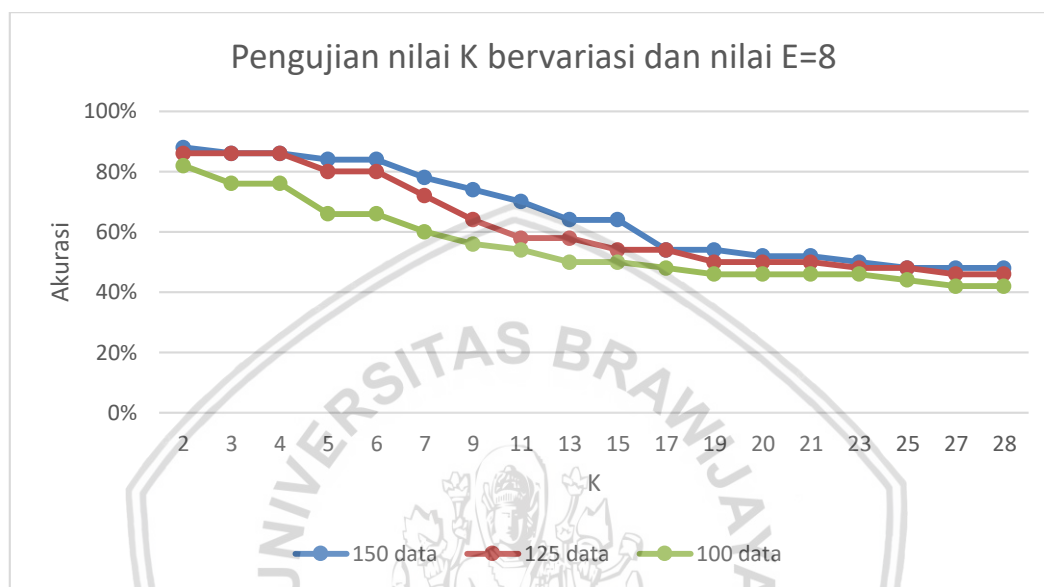
Pengujian dengan skenario ketiga yakni dilakukan dengan nilai E=8 dan nilai K berubah-ubah. Hasil nilai pengujian pertama dapat dilihat pada Tabel 6.4.

**Tabel 6.4 Hasil pengujian skenario 4 nilai E = 8**

Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
2	88%	86%	82%
3	86%	86%	76%
4	86%	86%	76%
5	84%	80%	66%
6	84%	80%	66%
7	78%	72%	60%
9	74%	64%	56%
11	70%	57.9%	54%
13	64%	57.9%	50%
15	64%	54%	50%
17	54%	54%	48%
19	54%	50%	46%
20	52%	50%	46%
21	52%	50%	46%



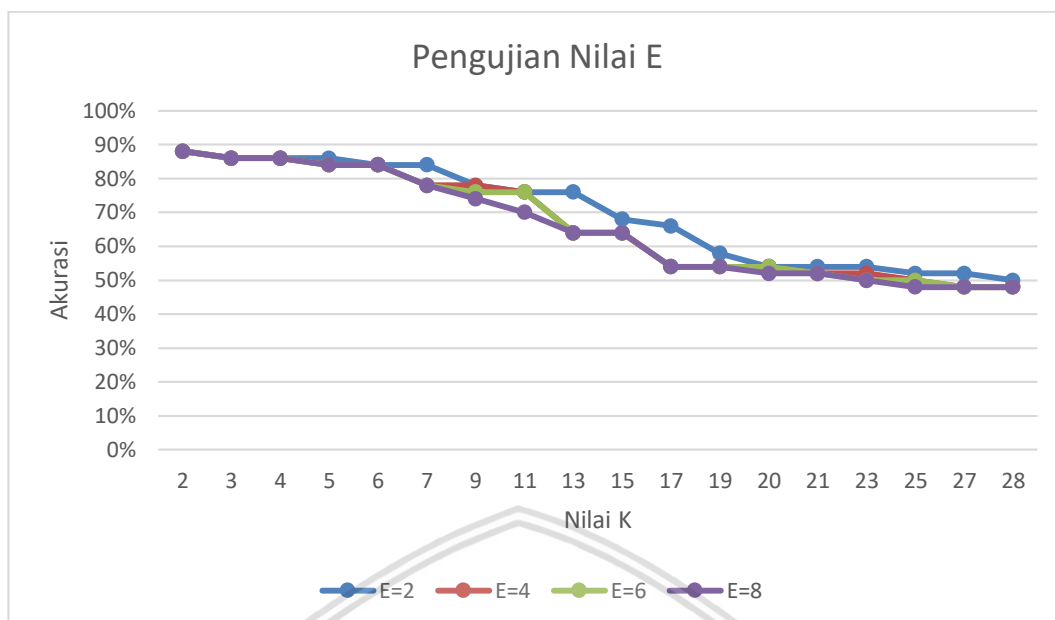
Nilai K	Tingkat Akurasi untuk Data Latih (%)		
	150 data	125 data	100 data
23	50%	48%	46%
25	48%	48%	44%
27	48%	46%	42%
28	48%	46%	42%



**Gambar 6.4 Grafik pengujian nilai K bervariasi dan nilai E=8**

Berdasar pada Tabel 6.4 dan Gambar 6.4, nilai akurasi sistem terbesar dengan nilai K=2, nilai E=8 dan komposisi data latih 150 adalah 88%. Pada pengujian skenario keempat ini sama seperti skenario sebelumnya yakni semakin besar nilai K pada semua komposisi data latih maka nilai akurasi semakin menurun. Hal ini dikarenakan hasil identifikasi kelas pada sistem memiliki perbedaan dengan kelas awal. Hasil identifikasi kelas yang berbeda ini terus bertambah seiring bertambahnya nilai K yang menjadikan tingkat akurasi semakin menurun. Kemudian komposisi data latih dengan 150 data memiliki nilai akurasi tertinggi dibandingkan komposisi data latih yang lain dikarenakan semakin banyak data latih yang digunakan, maka semakin banyak data yang dibandingkan dengan data uji dan memungkinkan untuk memperoleh hasil CosSim yang mempunyai similarity terbaik.

Dari keseluruhan skenario pengujian, maka diperoleh hasil gabungan pengujian pengaruh nilai E terhadap akurasi dengan menggunakan kombinasi data latih terbaik yakni 150 data. Hasil gabungan pengujian pengaruh nilai E terhadap akurasi dapat dilihat pada Gambar 6.5.



**Gambar 6.5 Grafik pengujian nilai E**

Berdasar dari yang telah disajikan pada Gambar 6.5 terlihat bahwa dengan semua nilai E pada nilai K=2, 3, 4 dan 6 memiliki nilai akurasi yang sama bagusnya yakni 88% pada K=2, 86% pada K=3, 4 dan 6, sedangkan untuk nilai K=5 dan K>6 memiliki nilai akurasi yang bervariasi dengan nilai E=2 terlihat unggul nilai akurasinya dari pada nilai E yang lain. Pada pengujian yang menggunakan data penyakit ginjal ini terlihat bahwa adanya perbedaan nilai akurasi dari perubahan nilai E. Nilai E tersebut terdapat pada perhitungan bobot setiap kelas, ketika nilai E diubah maka hasil dari bobot setiap kelas tersebut juga berubah. Hal ini menunjukkan bahwa perubahan nilai E serta nilai K yang digunakan pada penelitian ini memberikan pengaruh terhadap hasil akurasi pada sistem identifikasi penyakit gagal ginjal. Semakin besar nilai K pada semua nilai E, maka semakin turun pula tingkat akurasinya.

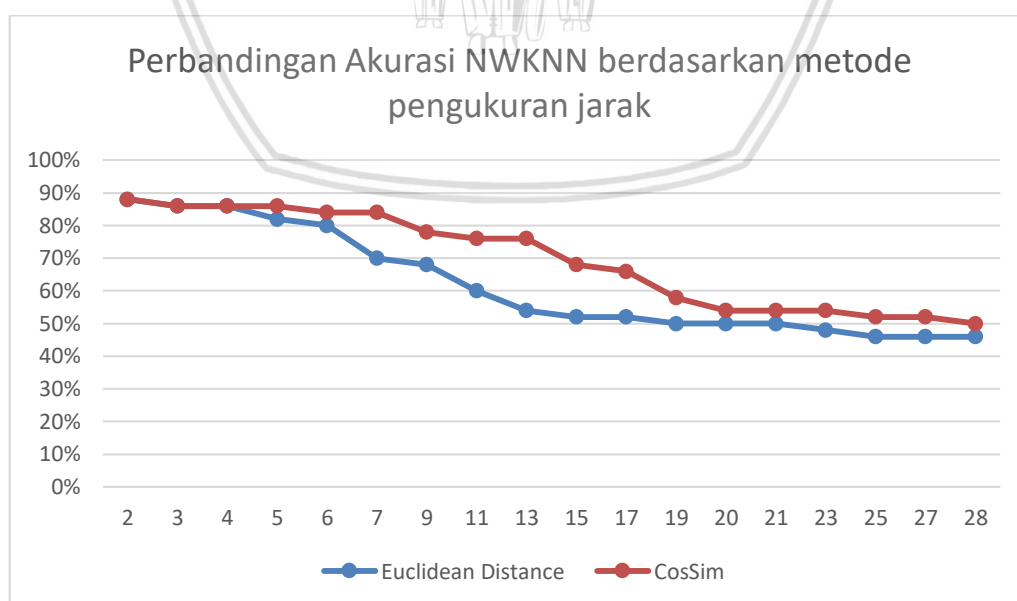
## 6.2 Pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak

Pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak menggunakan *Euclidean Distance* atau *CosSim* dilakukan untuk mengetahui bagaimana nilai akurasi pada sistem identifikasi penyakit gagal ginjal ketika perhitungan ketetanggaan terdekat pada NWKNN menggunakan *Euclidean Distance* atau *CosSim*. Pengujian ini dilakukan dengan menggunakan data latih sebanyak 150, data uji 50, nilai K yang bervariasi mulai dari 2 hingga 28, dan nilai E yang digunakan adalah nilai terbaik pada pengujian nilai E yakni bernilai 2.

Hasil pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak dapat dilihat pada Tabel 6.5 dan grafik perbandingannya dapat dilihat pada Gambar 6.6.

**Tabel 6.5 Hasil pengujian perbandingan akurasi metode NWKNN berdasarkan metode pengukuran jarak**

Nilai K	Tingkat Akurasi <i>Euclidean Distance</i>	Tingkat Akurasi <i>CosSim</i>
2	88%	88%
3	86%	86%
4	86%	86%
5	82%	86%
6	80%	84%
7	70%	84%
9	68%	78%
11	60%	76%
13	54%	76%
15	52%	68%
17	52%	66%
19	50%	57.9%
20	50%	54%
21	50%	54%
23	48%	54%
25	46%	52%
27	46%	52%
28	46%	50%
Hasil Rata-Rata Akurasi	62%	70%



**Gambar 6.6 Grafik hasil pengujian perbandingan akurasi NWKNN berdasarkan metode pengukuran jarak**

Berdasar Tabel 6.5 dan Gambar 6.6 terlihat bahwa perbandingan antara *Euclidean Distance* dan *CosSim* pada pengujian ini memberikan hasil bahwa NWKNN dengan menggunakan *CosSim* memiliki rata-rata nilai akurasi lebih baik 8% dibandingkan menggunakan *Euclidean Distance*. Ketika nilai  $K \leq 4$ , nilai akurasi *Euclidean Distance* dan *CosSim* sama dan terlihat bagus hasilnya ketika nilai  $K$  kecil. Tetapi ketika nilai  $K > 4$  maka perbandingan nilai akurasinya mulai terlihat. Meskipun nilai akurasi sama-sama turun ketika nilai  $K > 4$ , tetapi *CosSim* cukup baik dan konsisten dari pada *Euclidean Distance* ketika nilai  $K$  semakin besar. Ini menunjukkan bahwa perbandingan dari *Euclidean Distance* dan *CosSim* memiliki pengaruh terhadap hasil akurasi ketika nilai  $K$  semakin besar pada identifikasi gagal ginjal pada sistem.

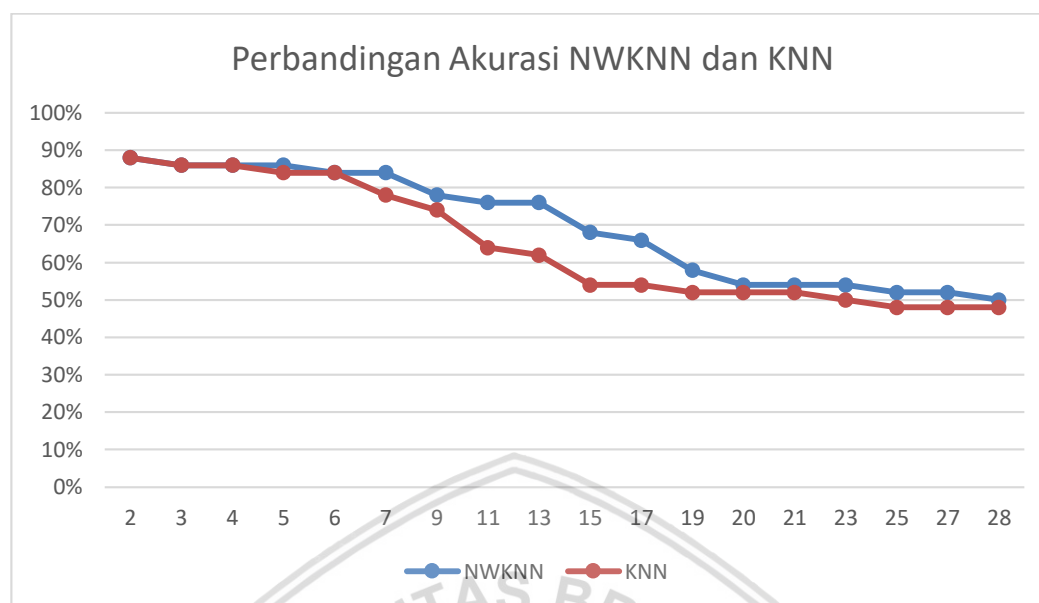
### 6.3 Pengujian perbandingan akurasi metode NWKNN dan KNN

Pengujian perbandingan akurasi metode NWKNN dengan KNN dilakukan untuk mengetahui bagaimana nilai akurasi pada sistem identifikasi penyakit gagal ginjal ketika menggunakan metode NWKNN dan KNN. Pengujian ini dilakukan dengan menggunakan data latih sebanyak 150, data uji 50, nilai  $K$  yang bervariasi mulai dari 2 hingga 28 dan kedekatan ketetanggaan menggunakan *CosSim*.

Hasil pengujian perbandingan akurasi metode NWKNN dan KNN dapat dilihat pada Tabel 6.6 dan grafik perbandingannya dapat dilihat pada Gambar 6.7 berikut ini.

**Tabel 6.6 Hasil pengujian perbandingan akurasi metode NWKNN dengan KNN**

Nilai K	Tingkat Akurasi NWKNN	Tingkat Akurasi KNN
2	88%	88%
3	86%	86%
4	86%	86%
5	86%	84%
6	84%	84%
7	84%	78%
9	78%	74%
11	76%	64%
13	76%	62%
15	68%	54%
17	66%	54%
19	57.9%	52%
20	54%	52%
21	54%	52%
23	54%	50%
25	52%	48%
27	52%	48%
28	50%	48%
Hasil Rata-Rata Akurasi	70%	65%



**Gambar 6.7 Grafik hasil perbandingan akurasi NWKNN dan KNN**

Berdasar Tabel 6.6 dan Gambar 6.7 terlihat bahwa perbandingan antara metode NWKNN dan KNN pada pengujian ini memberikan hasil bahwa metode NWKNN di beberapa nilai K lebih unggul dibandingkan metode KNN. Ketika nilai K 2, 3, 4, dan 6 nilai akurasi metode NWKNN sama dengan metode KNN, tetapi untuk nilai K yang lain metode NWKNN lebih unggul dibandingkan dengan metode KNN. Perbedaan nilai akurasi ini bisa saja terjadi karena metode NWKNN menggunakan bobot saat perhitungan skor, proses pembobotan ini dapat membantu identifikasi kelas yang berasal dari kelas minoritas pada data. Oleh karena itu, dapat dikatakan pada kasus identifikasi penyakit gagal ginjal dengan data yang tidak seimbang ini lebih baik menggunakan metode NWKNN dibandingkan metode KNN dengan hasil metode NWKNN lebih baik 5% rata-rata hasil akurasinya.

Pada penelitian ini tidak ada perbedaan bobot setiap atribut, contohnya atribut umur dan serum kreatinin memiliki bobot yang sama tetapi memiliki nilai atau isi yang berbeda. Oleh karena itu, metode NWKNN dan KNN memberikan akurasi sebesar 88%. Ketika bobot pada setiap atribut ditambahkan, mungkin akurasi yang didapatkan berubah menjadi naik atau turun.

## BAB 7 PENUTUP

Pada bagian ini akan dibahas terkait kesimpulan yang didapatkan terhadap penelitian yang dilakukan dan juga saran jika ada penelitian yang serupa atau pengembangan dari penelitian ini.

### 7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap identifikasi penyakit gagal ginjal menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN), dapat disimpulkan bahwa:

1. Cara mengidentifikasi gagal ginjal dengan menggunakan metode NWKNN adalah dengan cara memasukkan data pasien yang meliputi 24 gejala kemudian dilakukan normalisasi data menggunakan *min-max* (karena data memiliki selisih nilai yang besar). Selanjutnya menghitung nilai kedekatan ketetangga dengan menggunakan *Cosine Similarity* (*CosSim*) dengan menghitung nilai similarity antara data pasien terhadap data latih. Kemudian diurutkan berdasar nilai *CosSim* terbesar sesuai K tetangga terdekat. Selanjutnya dilakukan pembobotan sesuai kelas identifikasinya dan dihitung nilai skornya. Skor tertinggilah yang masuk kedalam identifikasi.
2. Pada pengujian parameter identifikasi penyakit gagal ginjal menggunakan metode NWKNN ini, nilai K yang digunakan adalah bernilai 2 hingga 28, nilai E yang digunakan bernilai 2, 4, 6 dan 8, serta kombinasi data latih yang digunakan adalah 150 data, 125 data dan 100 data. Nilai akurasi tertinggi yang didapatkan yakni sebesar 88% pada nilai K=2, dengan semua nilai E serta pada data latih sebanyak 150 data. Pada penelitian ini untuk nilai E yang terbaik adalah E dengan nilai 2 yang mana nilai E=2 lebih unggul nilai akurasinya pada beberapa nilai K dari pada nilai E yang lain.
3. Diketahui bahwa pada penelitian ini tingkat akurasi yang dihasilkan metode NWKNN mengalami peningkatan sebesar 5% dibandingkan metode KNN, dimana metode NWKNN memiliki rata-rata akurasi sebesar 70%, sedangkan metode KNN menghasilkan rata-rata akurasi sebesar 65%.

### 7.2 Saran

Saran yang dapat diberikan pada kelanjutan penelitian identifikasi penyakit gagal ginjal menggunakan metode NWKNN ini adalah untuk penelitian berikutnya diharapkan untuk mendapatkan nilai K atau nilai E tidak dicari secara manual atau dapat menemukan nilai K maupun nilai E secara optimal menggunakan metode optimasi supaya tidak terlalu lama untuk mendapatkan nilai K maupun nilai E yang terbaik.



## DAFTAR PUSTAKA

- Arissaputra, V., Ridok, A. & Indriati, 2015. *Klasifikasi Dokumen Tanaman Obat Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Malang: s.n.
- Budiyanto, M. A. K., 2002. *Gizi dan Kesehatan (Relevansi Gizi dengan Obesitas, Penyakit Ginjal, Penyakit Jantung, Diabetes Mellitus, Hipertensi, Kanker, Osteoporosis, Penyakit Saluran Cerna, Anemia, dan Penyakit pada Bayi dan Anak)*. Malang: Bayu Media dan UMM Press.
- Cahyaningsih, N. D., 2011. *Hemodialisis (Cuci Darah)*. Yogyakarta: Mitra Cendikia Press.
- Fadila, P. N., Indriati & Ratnawati, D. E., 2016. *Identifikasi Jenis Attention Deficit Hyperactivity Disorder (ADHD) Pada Anak Usia Dini Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Malang: s.n.
- Feizar, F. H., Indriati & Yusdistira, N., 2014. *Analisis Sentimen Opini Film Berbahasa Indonesia Berbasis Kamus Menggunakan Metode Neighbor Weighted K-Nearest Neighbor*. Malang: s.n.
- Indra, M. R., 2006. *Fisiologi Ginjal*. Malang: Fakultas Kedokteran Universitas Brawijaya.
- Jain, Y. K. & Bhandare, S. K., 2011. Min Max Normalization Based Data Perturbation Method for Privacy Protection. *International Journal of Computer & Communication Technology*, 11(8), pp. 45-50.
- Kementrian Kesehatan RI, 2013. *Riset Kesehatan Dasar 2013*. [Online] Available at: <http://www.depkes.go.id/resources/download/general/Hasil%20Risikesdas%202013.pdf> [Accessed 27 Januari 2017].
- Maghfirah, N., Dewi, C. & Hidayat, N., 2014. Implementasi Algoritma Fuzzy K-Nearest Neighbor (F-KNN) untuk Mengetahui Tingkat Resiko Penyakit Gagal Ginjal. *Doro: Repository Jurnal Mahasiswa Ptiik Universitas Brawijaya*, Volume 9.
- Ridok, A. & Latifah, R., 2015. Klasifikasi Teks Bahasa Indonesia Pada Corpus Tak Seimbang Menggunakan NWKNN. *Konfrensi Nasilnal Sistem & Informatika*, pp. 222-227.
- Rubini, L. & Eswaran, P., 2015. Generating comparative analysis of early stage prediction of Chronic Kidney Disease. *International Journal Of Modern Engineering Research (IJMER)*, V(7), pp. 49-55.
- Tan, S., 2005. Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, pp. 667-671.
- Waugh, A., Grant, A., Nurachmah, E. & Angriani, R., 2011. *Dasar-Dasar Anatomi dan Fisiologi*. Singapore: Elsevier.